



Universidad Carlos III de Madrid

PROYECTO FIN DE CARRERA

Ingeniería Técnica en Electrónica y Automática

PREPARACIÓN DE PRÁCTICAS SIMULADAS PARA LA AUTOMATIZACIÓN INDUSTRIAL

TUTORA: MARÍA ANGELES MALFAZ VÁZQUEZ

María Lourdes Ruiz Sánchez
Leganés, Octubre 2015

AGRADECIMIENTOS.

Gracias, quiero comenzar dando gracias a Dios por mi vida y por la oportunidad que me ha dado de poder estudiar naciendo aquí y ahora. Por todo lo que cada día me regala. Dones pero sobre todo por las personas que me han ayudado a crecer, a ser como soy.

Doy gracias por mis padres, a quienes dedico este proyecto. GRACIAS POR TODO. Vosotros lo sabéis, por todo. Os quiero.

Gracias a mis hermanos. Víctor y Carlos. Gracias por ser mis compañeros de juegos, por todas esas risas y juegos que tanto echo de menos. ¿Por qué vivís tan lejos? Os quiero.

Gracias a mi familia, en especial a mis abuelos. Cómo os echo de menos, a la tía, no puedo pensar en ella sin llorar, cuanto te quise, cuanto me enseñaste, a querer a todos sin importar nada. Qué ejemplo, ojalá algún día me parezca a ti. Al abuelo, cuanto nos quisimos, cuantas trastadas juntos... A Toñín y a la Maru, os mando besos al Cielo, gracias a todos por creer en mí. Por apoyarme siempre, con mis dolores... confiando siempre en mí.

Llegados a este momento de la vida gracias a mi nueva familia. A Alfonso, amor, cuanto me has ayudado especialmente estos últimos años con tantos dolores y con mi salud tan perjudicada. Siempre creíste en mí, y mira estoy escribiendo los agradecimientos de ese proyecto que tú siempre supiste que haría. Gracias a nuestros dos soles, Clara y María, lo mejor que nos ha pasado. Perdonadme por el tiempo que os he robado haciendo este proyecto. Os quiero mis princesas.

Por último, porque lo mejor es para el final, gracias a ti María. Por ser y estar. Siempre cuidándome y ayudándome, especialmente en este momento tan difícil de salud. Por darme siempre tu apoyo y hacerme sentir bien. Eres un súper regalo de Dios, junto a Antonio al que le debo conocerte.

¡Gracias a todos!

INDICE DE CONTENIDOS:

AGRADECIMIENTOS.	2
CAPITULO 1.	4
MOTIVACIÓN Y OBJETIVOS	4
1.1 MOTIVACIÓN.	4
1.2 OBJETIVOS.	4
1.3 ESTRUCTURA DEL DOCUMENTO	4
CAPITULO 2.	5
INTRODUCCIÓN AL	5
SOFTWARE UNITY-PRO SCHNEIDER	5
2.1 DESCRIPCIÓN.	5
2.2 LENGUAJES DE PROGRAMACIÓN	6
2.3 SIMULADOR DE UNITY PRO	7
CAPITULO 3.	8
EJERCICIO DESARROLLADO	8
3.1 ENUNCIADO DEL EJERCICIO PROPUESTO	8
3.2 CONFIGURACIÓN DEL ENTORNO DE TRABAJO CON UNITY PRO	9
3.3 VARIABLES	11
3.4 PROGRAMACIÓN DE LA SOLUCIÓN DEL EJERCICIO. DIAGRAMAS SFC.	13
CAPITULO 4.	18
SIMULACIÓN DEL EJERCICIO.	18
PANTALLA DEL OPERADOR.	18
4.1 CREACIÓN DE LAS PANTALLAS DEL OPERADOR.	18
4.2 CONFIGURACIÓN DEL EJERCICIO DESARROLLADO	21
4.3 EJECUCIÓN DEL PROGRAMA	26
4.4 EVOLUCIÓN DEL SISTEMA. BOTONES DE COMANDO	29
CAPITULO 5.	30
MODO DOCENTE EN EL LABORATORIO.	30
CAPITULO 6.	33
BIBLIOGRAFÍA	33
CAPITULO 7.	34
ANEXOS (DOCUMENTACIÓN TÉCNICA)	34

CAPITULO 1

Motivación y objetivos

1.1 Motivación.

El presente proyecto surge de la necesidad de aprovechar las aplicaciones que nos ofrece el nuevo software de Schneider presente en los laboratorios de Automatización Industrial de la Universidad. El software Unity Pro que, a diferencia de su predecesor, el PL7, tiene la posibilidad de simular un proceso industrial fuera del laboratorio sin necesidad de estar físicamente conectados a un autómatas.

Por este motivo, este proyecto presenta la descripción del desarrollo de la simulación de un ejercicio típico de automatización de procesos industriales, así como la preparación de material docente para su utilización en la asignatura de Automatización Industrial.

1.2 Objetivos.

De acuerdo con lo anteriormente descrito, el objetivo de este proyecto es la preparación de material docente para su uso en la asignatura de Automatización Industrial. Es decir, los alumnos recibirán la pantalla de simulación gráfica (pantalla de operador) con los elementos propios del ejercicio además de las variables necesarias para su correcta programación. El objetivo de los alumnos será programar en Diagrama Funcional Secuencial (SFC), utilizando las variables dadas, y conseguir la simulación gráfica funcione acorde con el enunciado.

Para ello se necesitan cumplir los siguientes sub-objetivos:

- Familiarizarse con el software Unity Pro (programación en SFC y programación de la simulación de procesos industriales)
- Programar en SFC el ejercicio propuesto
- Programar la simulación de los elementos gráficos necesarios para del ejercicio.
- Preparación del material docente que se entregará a los alumnos.

1.3 Estructura del documento

Este documento se estructura de la siguiente manera:

- En el Capítulo 2 se hará una breve introducción al software Unity Pro.
- El Capítulo 3 se presenta el ejercicio a programar y simular.
- El Capítulo 4 describe de manera detallada la simulación gráfica del funcionamiento del proceso industrial propuesto.
- En el Capítulo 5 se presenta el material docente preparado.
- El Capítulo 6 presenta la Bibliografía y los anexos con la documentación generada por el software.

CAPITULO 2

Introducción al Software Unity-Pro Schneider

2.1 Descripción

El software de programación Unity Pro XL v.6 [1], es un software de última generación y de reciente implantación en el campo de la automatización industrial. Ante tal eventualidad, se va a entrar en detalles sobre su modo de operación en el capítulo 4.

Dentro del software Unity Pro XL podemos generar proyectos en varias familias de autómatas, como son Modicom M340, Premium, Quantum y Atrium. Todos ellos dentro de la marca Schneider Electric.

Las características del software industrial son las mencionadas a continuación:

- Está basado en formatos estándares, reduciendo sensiblemente las actividades de familiarización y capacitación.
- Presenta las herramientas necesarias para la creación, depuración y puesta en marcha de las aplicaciones.
- El entorno de ejecución de los programas es Windows 98/2000/NT/XP/7, estando adaptado a su funcionamiento gráfico y orientado a objetos.
- Su utilización es intuitiva.

Unity Pro aprovecha al máximo las ventajas de los interfaces gráficos y contextuales de Windows XP y Windows 2000:

- Acceso directo a las herramientas y a los datos.
- Configuración 100% gráfica.
- Barra de herramientas e iconos personalizables.
- Funciones avanzadas de “arrastrar y soltar” y zoom.
- Ventana de diagnóstico integrado

La principal ventaja de este software de programación es la estandarización de sus códigos gracias al cumplimiento de la norma IEC 61131-3 [2], con la cual se consigue que Unity Pro proponga un conjunto completo de funcionalidades y de herramientas que permiten calcar la aplicación en la estructura del proceso o de la máquina.

2.2 Lenguajes de programación

La norma IEC 61131-3 [2] persigue la estandarización de los lenguajes en la programación del control industrial. Bajo esta norma, se definen cuatro lenguajes de programación normalizados. Esto significa que su sintaxis y semántica ha sido definida, no permitiendo particularidades distintivas (dialectos).

Los lenguajes consisten en dos de tipo literal y dos de tipo gráfico:

Literales:

- Lista de instrucciones ("*Instruction List*", "*IL*"): es el modelo de lenguaje ensamblador basado en un acumulador simple; procede del alemán *Anweisungsliste*, *AWL*.
- Texto estructurado ("*Structured Text*", "*ST*"): es un lenguaje de alto nivel con orígenes en ADA, Pascal, y 'C'; puede ser utilizado para codificar expresiones complejas e instrucciones anidadas; este lenguaje dispone de estructuras para bucles (*REPEAT-UNTIL*; *WHILE-DO*), ejecución condicional (*IF-THEN-ELSE*; *CASE*), funciones (*SQRT*, *SIN*, etc.)

Gráficos:

- Diagrama de contactos ("*Diagram Ladder*", "*LD*"): está basado en la presentación gráfica de la lógica de relés.
- Diagrama de bloques funcionales ("*Function Block Diagram*", "*FBD*"): es muy común en aplicaciones que implican flujo de información o datos entre componentes de control. Las funciones y bloques funcionales aparecen como circuitos integrados y es ampliamente utilizado en Europa.

Adicionalmente, el estándar *IEC 1131-3* incluye una forma de programación orientada a objetos llamada *Sequential Function Chart (SFC)*. *SFC* es a menudo categorizado como un lenguaje *IEC 1131-3*, pero éste es realmente una estructura organizacional que coordina los cuatro lenguajes estándares de programación (*LD*, *FBD*, *IL* y *ST*). La estructura del *SFC* tuvo sus raíces en el primer estándar francés de *Grafcet (IEC 848)*. Es un "lenguaje" gráfico que provee una representación diagramática de secuencias de control en un programa. Básicamente, *SFC* es similar a un diagrama de flujo, en el que se puede organizar los subprogramas o subrutinas (programadas en *LD*, *FBD*, *IL* y/o *ST*) que forman el programa de control. *SFC* es particularmente útil para operaciones de control secuencial, donde un programa fluye de un punto a otro una vez que una condición ha sido satisfecha (cierta o falsa).

El marco de programación de *SFC* contiene dos principales elementos que organizan el programa de control:

- Pasos (etapas)
- Transiciones (condiciones)

El programa irá activando cada una de las etapas y desactivando la anterior conforme se vayan cumpliendo cada una de las condiciones. Las acciones se realizarán en función de la etapa activa a la que están asociadas.

2.3 Simulador de Unity Pro

Una novedad dentro del software de programación de Schneider Electric es un simulador del autómatas integrado. Reproduce fielmente el comportamiento del programa en el PC. Gran parte de las herramientas de puesta a punto se pueden utilizar en simulación para aumentar la calidad antes de la instalación. Por tanto, gracias a este simulador se puede llevar el código pre-depurado a la instalación, con lo cual se reduce el tiempo de puesta en marcha del proceso [3]. Entre las funciones que se pueden desarrollar con el simulador, destaca:

- Ejecución del programa paso a paso.
- Punto de parada y de visualización
- Animaciones dinámicas para visualizar el estado de las variables y la lógica que se está ejecutando.

CAPITULO 3

Ejercicio Desarrollado

3.1 Enunciado del ejercicio propuesto

Tenemos un entorno de fabricación constituido por dos cintas transportadoras, un robot y un palet según se representa en la Figura 1. El autómatas controla el sistema que evoluciona de la siguiente manera:

- Pulsando un botón de arranque se inicia el proceso.
- Por la cinta 1 llegan cajas. El detector de proximidad detecta una caja y el autómatas para la cinta.
- El autómatas activa el programa del robot.
- Cuando el robot ha colocado una caja en el palet, este manda una señal al autómatas y la cinta 1 se vuelve a poner en funcionamiento.
- Cuando el robot ha colocado 4 cajas en el palet, el autómatas activa la cinta 2 durante 5 segundos.
- Al final, todo vuelve a la posición de reposo.

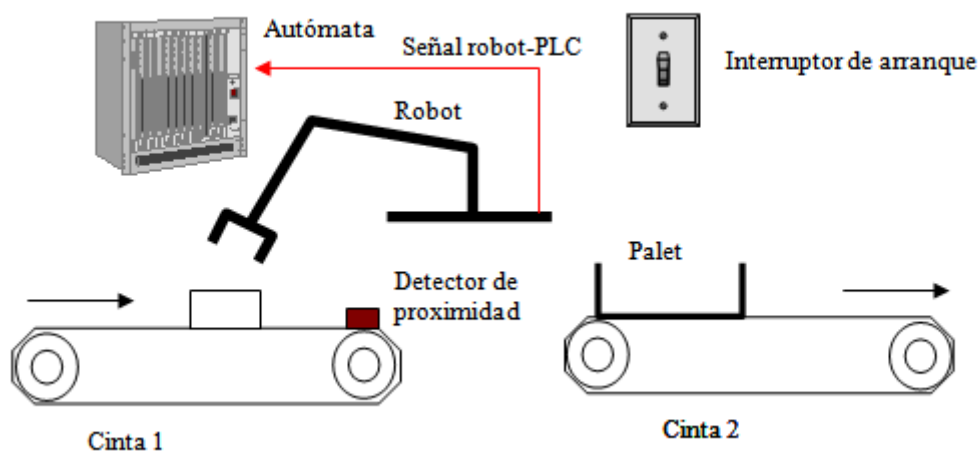


Figura 1: Ejercicio propuesto

Se pide programar utilizando un Diagrama Funcional Secuencial (SFC) el funcionamiento del sistema descrito previamente.

El objetivo de esta parte es mostrar y explicar detalladamente la programación en SFC y la simulación de este ejercicio utilizando el software Unity Pro.

Con el fin de simular el proceso, dividimos el enunciado en partes para realizar la simulación.

Inicio del proceso:

El proceso comenzará cuando pulsemos el botón de arranque, IA.

Cinta 1:

La cinta número 1 se activa cuando iniciamos el modo de producción activando el botón IA y se para cuando la caja haya llegado al final de la cinta. Este momento se detecta mediante del sensor de presencia simulado con el botón DP.

Robot:

El robot se activa cuando la caja llega a la posición detectada por el sensor DP. El robot recoge la caja de la cinta 1 y la lleva al palet que la recoge en la cinta 2.

Cinta 2:

La cinta 2 se pone en movimiento cuando el palet está completo (tiene 4 cajas) y se mueve durante 5 segundos.

3.2 Configuración del entorno de trabajo con Unity Pro

Para comenzar a programar debemos elegir el modelo de autómeta con el que vamos a trabajar. De las diferentes familias de PLC's que tiene Schneider Electric utilizaremos la Premium. Esta es la familia disponible en el laboratorio de Automatización Industrial.

Abrimos el programa Unity Pro para crear un nuevo proyecto.



Pulsando sobre el icono anterior se nos abre el programa. Seleccionamos Fichero, Nuevo y tenemos un cuadro como el mostrado en Figura 2. Ahí hemos de seleccionar de la familia indicada el modelo: **TSX P57 104M 02.90 57-1**.

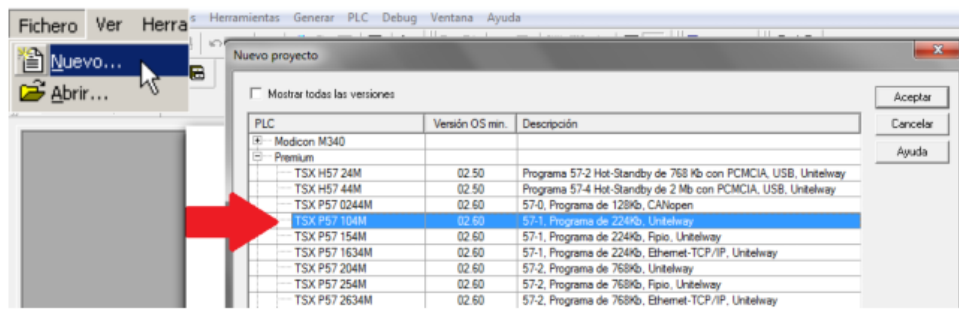


Figura 2: Ventana de Nuevo Proyecto

Así tenemos el siguiente entorno de trabajo, ver Figura 3:

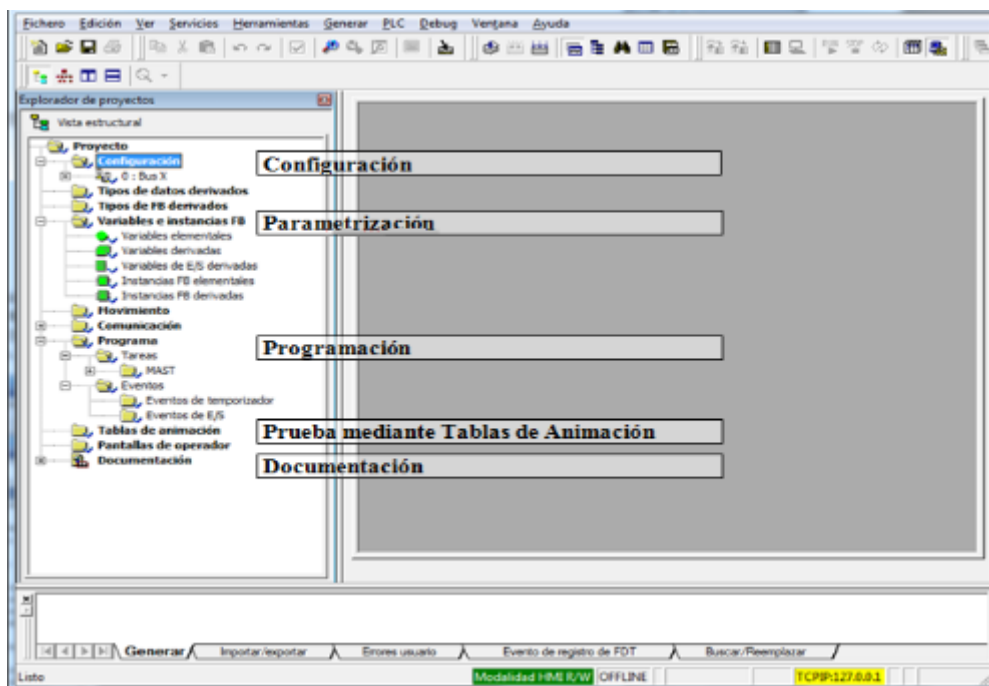


Figura 3: Explorador de proyectos

Desde esta pantalla en el Explorador de Proyectos hemos de configurar el Hardware con el que vamos a trabajar. Ha de ser el mismo que el que tendrán los alumnos en el laboratorio.

Desde esta pantalla en el Explorador de Proyectos hemos de configurar el Hardware con el que vamos a trabajar. Ha de ser el mismo que el que tendrán los alumnos en el laboratorio.

Hemos de instalar los módulos en el rack donde está la CPU y la fuente de alimentación instalada automáticamente por el programa. En la siguiente figura, ver Figura 4, podemos observar que el rack está dividido en varios slots. La posición 0 está ocupada por la CPU y a continuación todos los módulos que necesitamos.

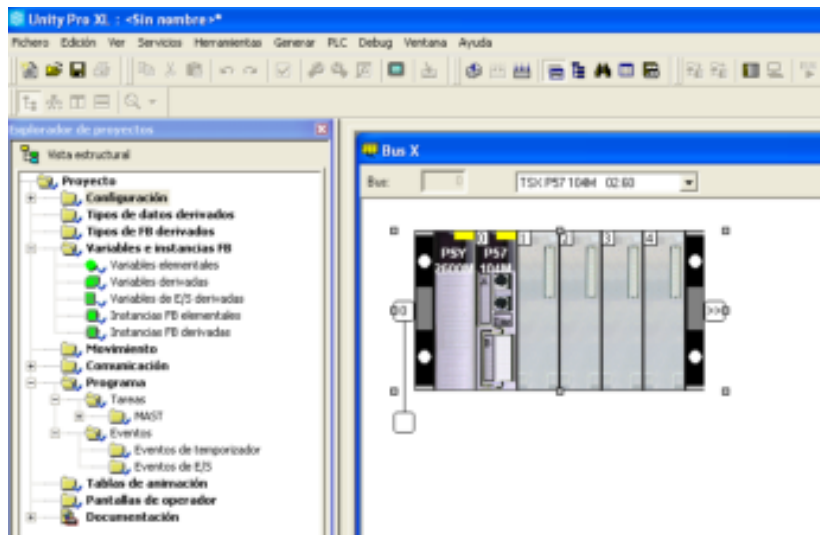


Figura 4: Configuración hardware

Una vez seleccionado el tipo de rack del que disponemos (un rack de 8 posiciones), vemos los diferentes módulos que forman parte de nuestro rack, ver Figura 5.

El primero con las siglas PSY es el correspondiente a la fuente de alimentación de 100/240 VCA 26W.

El segundo con la denominación P57 es el procesador de la CPU.

El tercer bloque tiene las entradas digitales binarias con la denominación DEY 16FK.

Las salidas digitales están en el cuarto bloque, en el módulo con la denominación DSY 16R5.

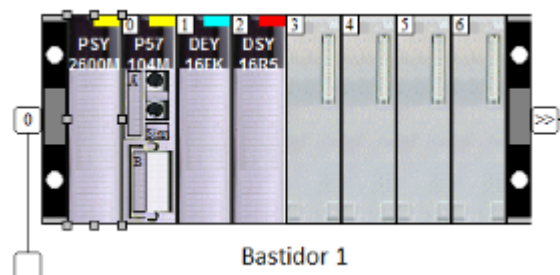


Figura 5: Módulos del autómata

Esta configuración inicial es suficiente para el ejercicio que vamos a realizar pero es modificable. Podríamos añadir un rack mayor, un segundo rack, módulos tanto binarios como analógicos, módulos de conteo o de comunicación.

3.3 Variables

Las variables que hemos utilizado para la realización de este ejercicio se pueden ver en el Explorador de Proyectos dentro de la sección Variables e Instancias FB.

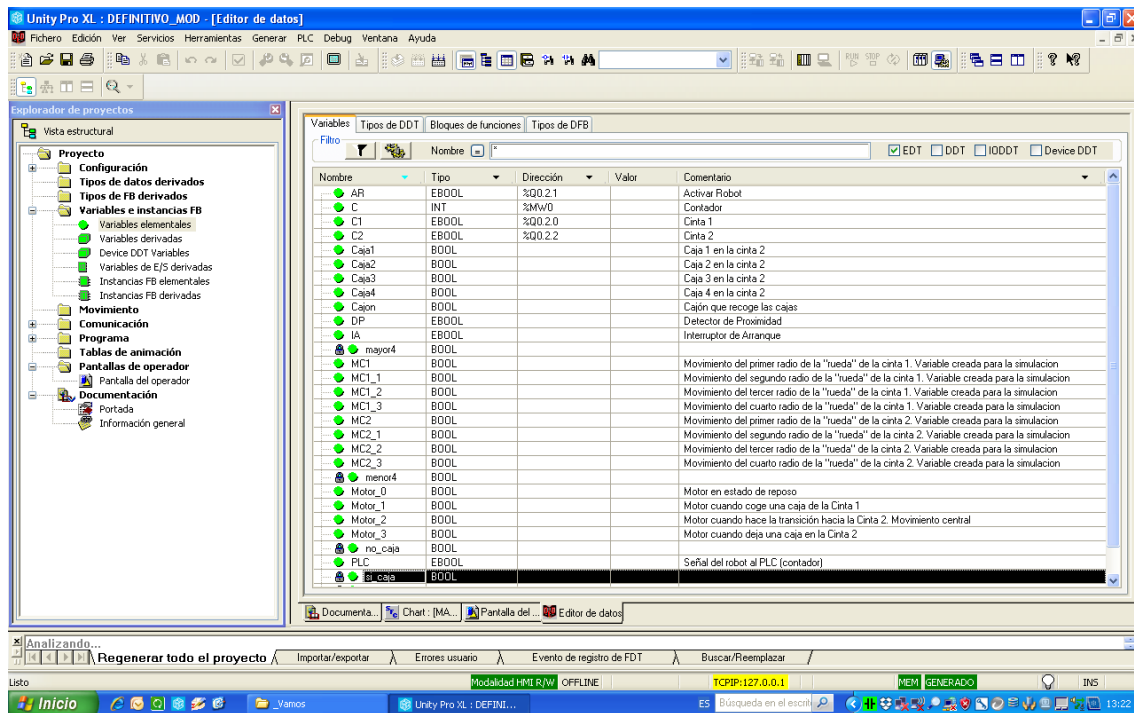


Figura 6: Tabla de variables

Las variables mostradas en la Figura 6 pueden ser clasificadas como: variables del programa o variables de simulación.

Las *variables del programa* son usadas para la programación en SFC del ejercicio propuesto. Estas variables son las correspondientes a las entradas y salidas del sistema.

Las *variables de simulación* han sido creadas solamente para la simulación gráfica, estas no se utilizan en el código de programación del ejercicio.

Para definir la variable hemos de escribir los siguientes parámetros:

Nombre, tipo, dirección, valor y comentario.

Por último, las variables que tienen un candado a la izquierda, las variables derivadas, ver Figura 7, son generadas automáticamente por el software, y corresponden a variables ligadas a las etapas de los SFCs.

Los contadores son de tipo entero y los temporizadores son utilizados para contar el tiempo. En nuestro caso, los 5 segundos.

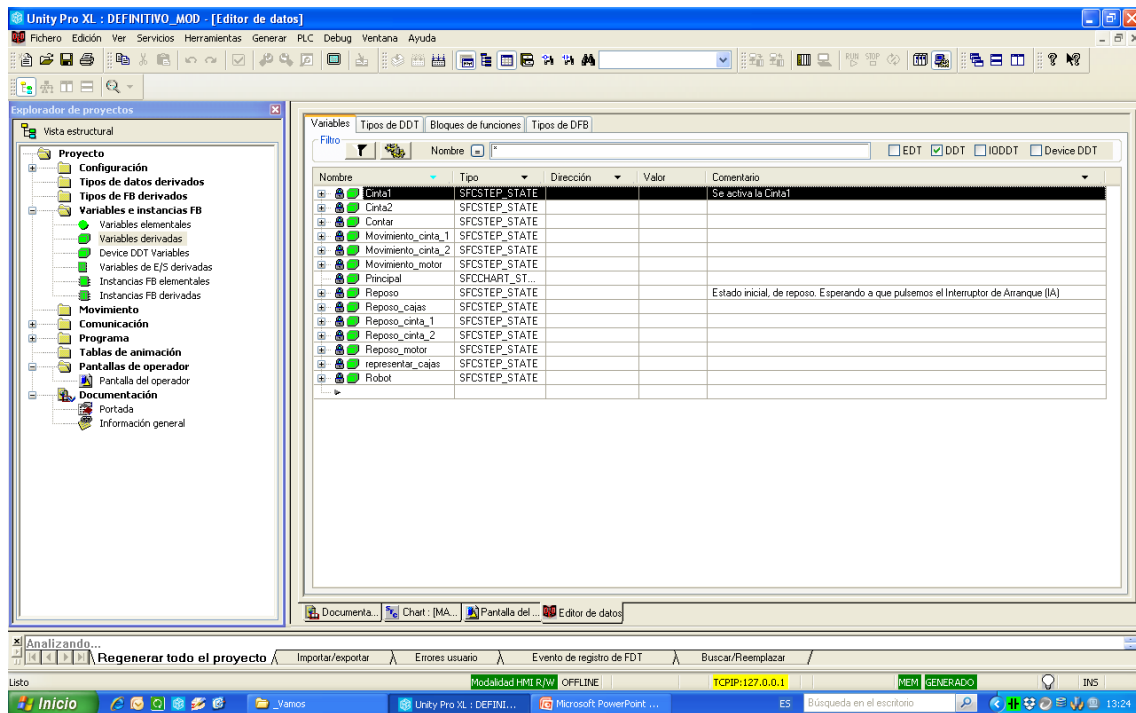


Figura 7: Variables derivadas

3.4 Programación de la solución del ejercicio. Diagramas SFC

En este apartado vamos a realizar la programación del problema automatizado mediante un diagrama SFC, programado en LD.

Para definir el diagrama SFC hemos de crear una nueva sección SFC, ver Figura 8:

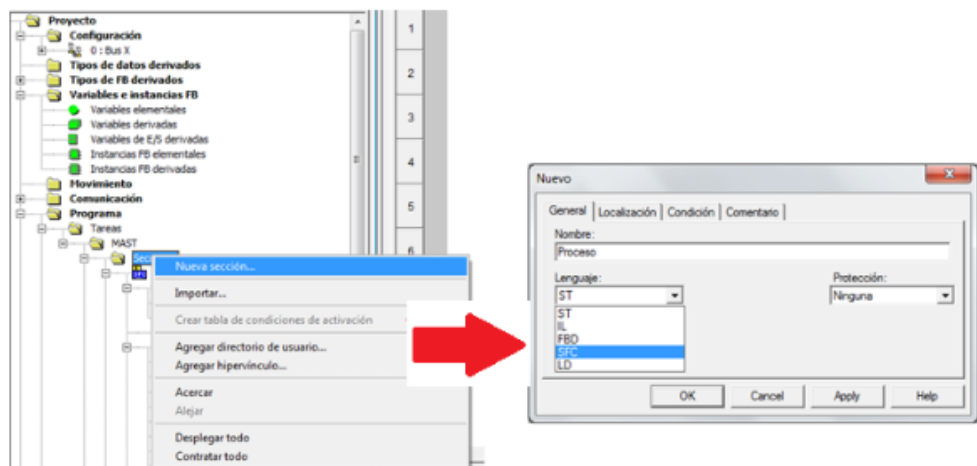


Figura 8: Selección del lenguaje de programación

Al elegir realizar una representación SFC en Unity Pro, el software nos ofrece la siguiente pantalla, Figura 9.

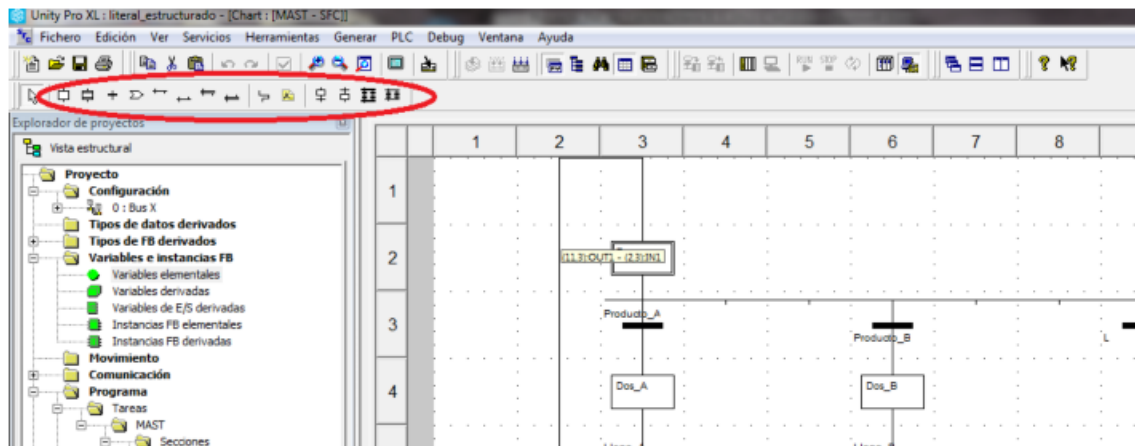


Figura 9: Pantalla SFC

Los elementos necesarios para crear un diagrama SFC están en la barra de herramientas mostrada en la Figura 10:



Figura 10: Barra de elementos de programación de SFC

Para nuestro ejercicio propuesto el diagrama SFC final es el que se ve en la Figura 11. Este diagrama unido a los siguientes, creados solo para la simulación, harán posible que el alumno vea simulado mediante gráficos la ejecución de su sistema automatizado. La programación detallada de cada una de las etapas y transiciones de estos SFCs se muestran en el anexo de este proyecto.

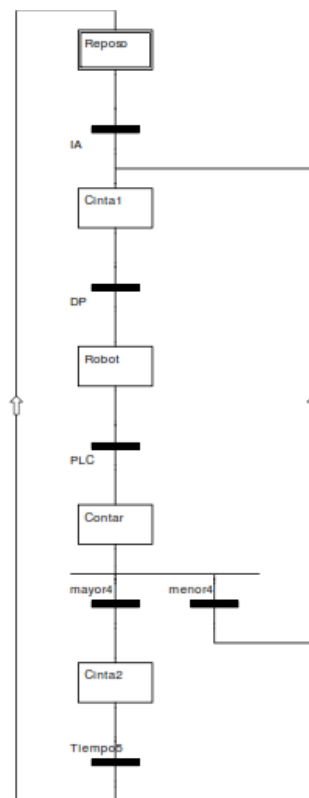
SFC para el control del proceso:

Figura 11: SFC de la programación del ejercicio propuesto

Este diagrama principal es el que el alumno ha de desarrollar. En él se gestiona todo el proceso. Recibe la señal IA, de inicio del proceso. Activa la cinta 1 hasta que se detecta por el sensor DP que hemos llegado al final de la misma. Manda la señal de activación al robot. Mediante la señal PLC el autómatas cuenta el número de cajas y discierne si hemos llegado al valor máximo poniendo así en movimiento la cinta número 2 durante 5 segundos.

Programación de etapas:

- Etapa Cinta 1: Activar variable C1 (descriptor N)
- Etapa Robot: Activar variable AR (descriptor N)
- Etapa Contar: Aumentar en 1 la variable INT C (sección InC descriptor P1)
- Etapa Cinta 2:
 - Activar variable C2 (descriptor N)
 - Poner a cero la variable INT C (sección cont_0 descriptor P1)

Programación de las transiciones:

- Transición IA: se activa al activar la variable IA
- Transición DP: se activa al activar la variable DP
- Transición PLC: se activa al activar la variable PLC
- Transición mayor4: se activa cuando la variable C es mayor que 4 (sección)
- Transición menor4: se activa cuando la variable C es menor que 4 (sección)

- Transición Tiempo5: se activa cuando la etapa Cinta 2 lleva activa 5 segundos (sección)

SFC's para controlar la simulación:

Tal y como se observa en la Figura 12 y Figura 13, para el movimiento de las cintas 1 y 2, el movimiento del robot y el dibujo de las cajas en el palet hemos utilizado otros esquemas SFC para la simulación. Todos estos diagramas han sido programados con el lenguaje LD. En Capítulo 4, se darán mas detalles de su programación.

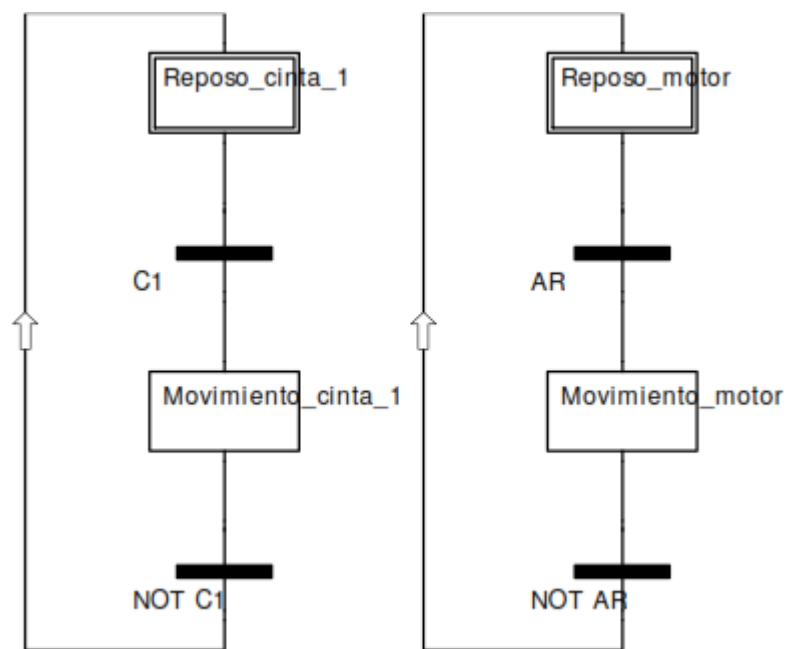


Figura 12: SFCs que controlan el movimiento de la cinta 1 y del robot

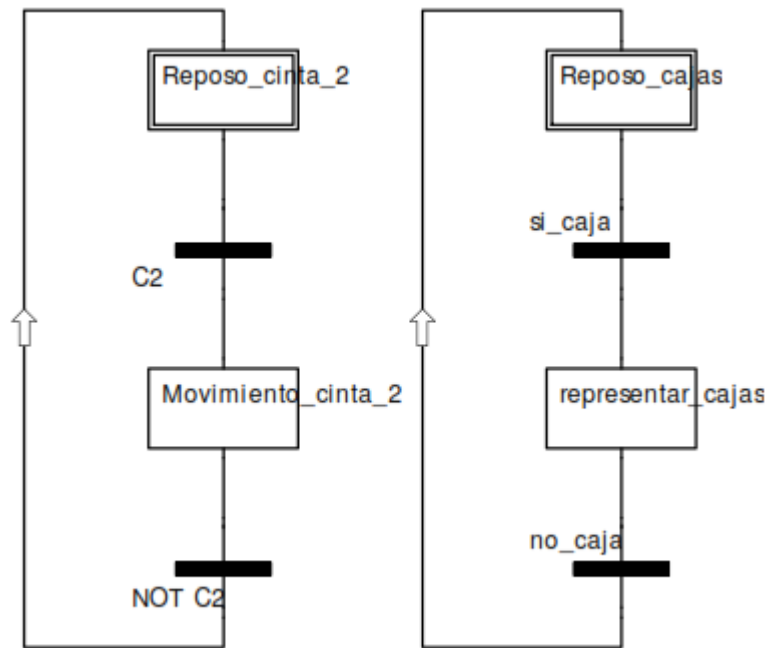


Figura 13: SFCs que controlan el movimiento de la cinta 2 y la representación de las cajas en el palet.

CAPITULO 4

Simulación del ejercicio.

Pantalla del operador.

4.1 Creación de las Pantallas del Operador

Para ver gráficamente la evolución del ejercicio simulado, Unity Pro nos ofrece la posibilidad de visualizar gráficamente los procesos cuando tenemos funcionando el simulador trabajando de manera offline.

Primero hemos de crear el proyecto y programar el proceso que queremos simular, después crearemos una imagen gráfica del proceso programado.

Los elementos gráficos los podemos crear mediante la Pantalla del Operador. Desde esta pantalla, el usuario puede utilizar los gráficos que nos ofrece Unity Pro en su librería de pantallas del operador, o bien insertar imágenes e incluso crear dibujos. Así podremos usar botones, barras, cintas, bombas, tanques, etc. o dibujar formas según nuestra necesidad. Estos elementos pueden ser estáticos o dinámicos.

Para crear la Pantalla del Operador hemos de ir la anteriormente mencionada ventana “Explorador de Proyecto”, con el botón derecho pulsamos sobre “Pantallas de Operador” y nueva pantalla, como se ve en la Figura 14:

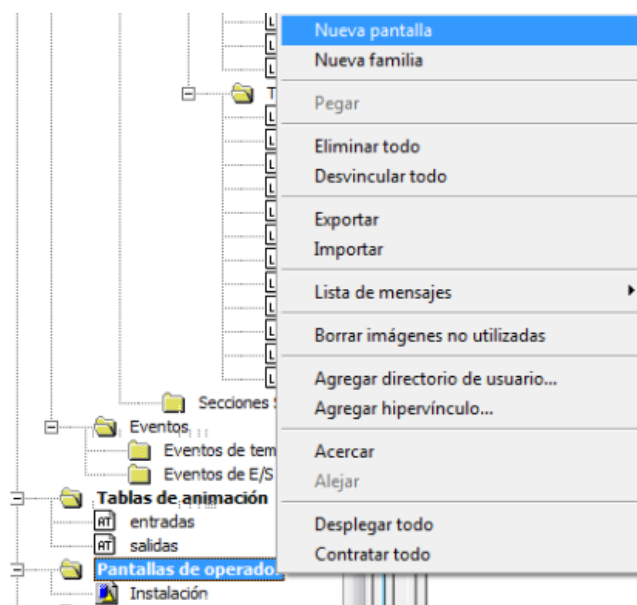


Figura 14: Creación pantalla de operador

Así nos aparecerá una pantalla en blanco lista para poner los elementos necesarios. El propio programa ofrece muchos elementos ya diseñados. Para utilizarlos hemos de ir a Herramientas y elegir la opción de Librería de Pantallas de operador.

Nos aparecerá una ventana como la mostrada en la Figura 15. Los elementos vienen agrupados por familias.

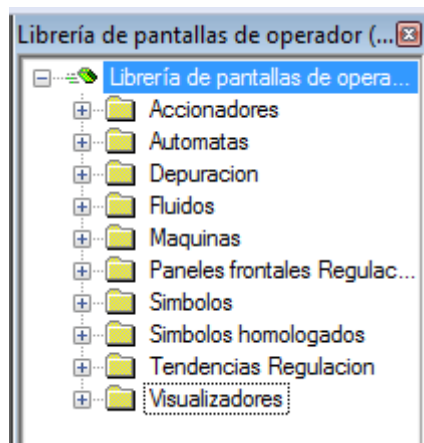


Figura 15: Librería de pantallas de operador

De esta librería hemos utilizado las cintas transportadoras de nuestro ejercicio, ver Figura 16.

Máquinas (Cinta transportadora)

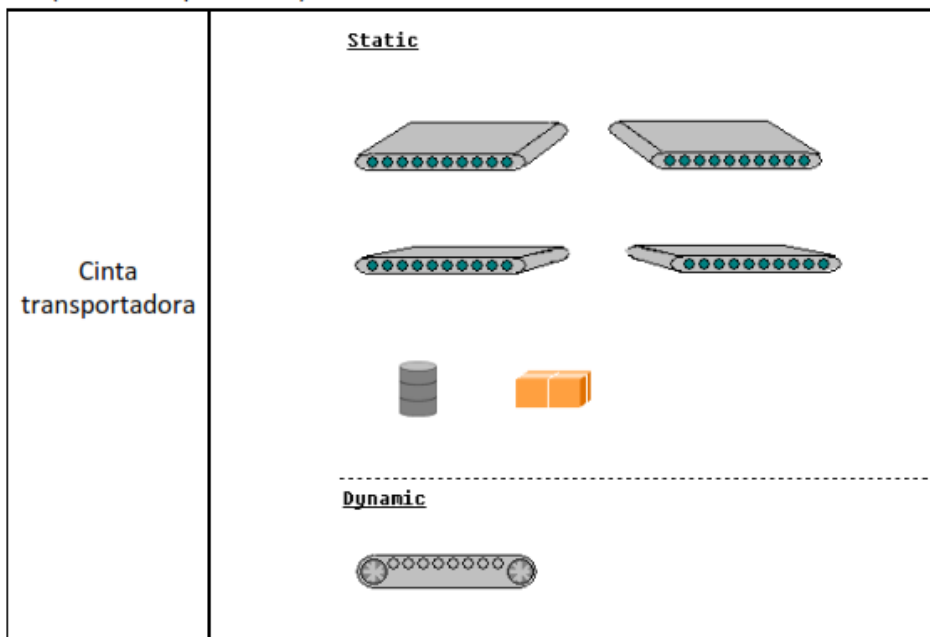


Figura 16: Cintas transportadoras

A continuación, se presenta la clasificación de los elementos gráficos según el movimiento que se transmita:

- Los ***elementos estáticos***, como su nombre indica, aparecen en la simulación representando a un objeto concreto pero no ofrece ningún movimiento en su simulación. Puede aparecer durante todo el tiempo de la duración de la simulación, o si están asociados a una variable booleana, el máximo “dinamismo” (no olvidando que se habla de elemento estático), que puede ofrecer, es que aparezca cuando esté activa la variable y desaparezca cuando esté inactiva, o viceversa.
- Los ***elementos dinámicos asociados a variable booleana***, producen movimiento en el transcurso de la simulación. El dinamismo, va a ir en función del objeto. En nuestro caso, la cinta transportadora dinámica tiene elementos dinámicos, que son las aspas, para poder representar su movimiento de avance o retroceso.
- Los ***elementos dinámicos con variables tipo entero (INT)*** asociada, van a producir un dinamismo más real. En el caso de un nivel de un tanque el movimiento es la subida del nivel de capacidad del depósito.

Los elementos son totalmente combinables entre ellos y modificables. El programa nos ofrece la posibilidad de crear otros como hemos comentado mediante la siguiente barra de herramientas, Figura 17:

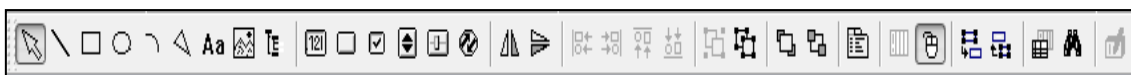


Figura 17: Barra de herramientas

4.2 Configuración del ejercicio desarrollado

El aspecto que tiene nuestra instalación en uno de los momentos de la simulación se muestra en la Figura 18:

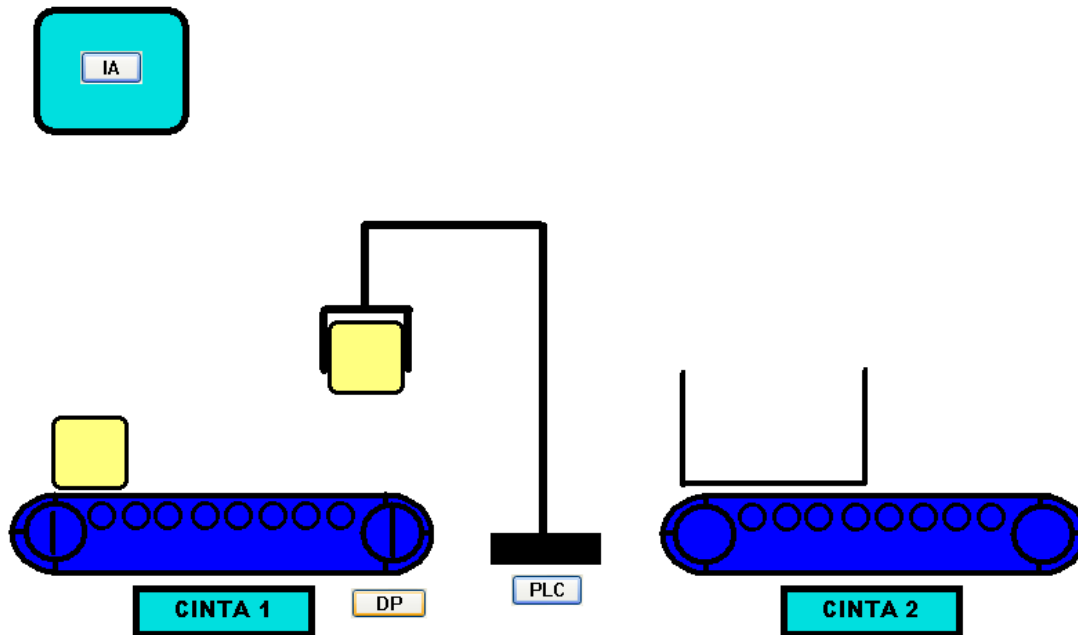


Figura 18: Una escena de la simulación

Tal y como se observa, los sensores del proceso (las entradas para el autómata) están representadas por botones. De esta manera, se controla la simulación directamente pulsando cada uno de ellos.

Una vez seleccionados los elementos que necesitamos para ilustrar nuestra instalación hemos de ir animándolos. Los elementos no se animan de forma conjunta, sino que hay que animarlo por partes.

En este apartado vamos a describir los elementos utilizados:

Botones de comando:

Para introducir los botones de comando necesitamos seleccionar el elemento de la barra de herramientas, tal y como se muestra en la Figura 19.

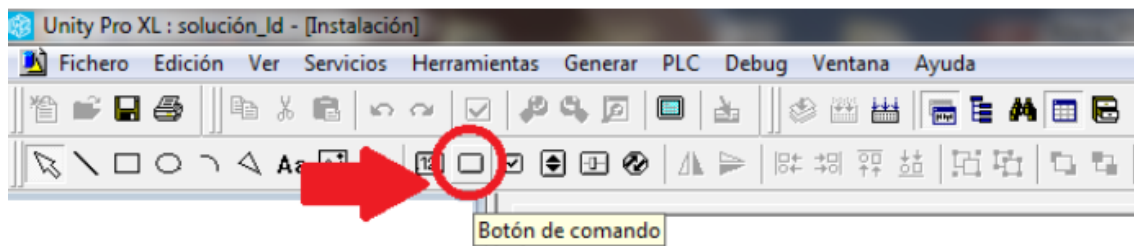


Figura 19: Selección del botón de comando

Una vez seleccionado, se debe elegir el punto en la pantalla de operador donde lo queremos situar y después debemos arrastrar el ratón con el botón izquierdo mantenido para determinar el tamaño del botón.

Posteriormente, se determina el nombre, la variable que activa y su funcionamiento, ver

Figura 20:

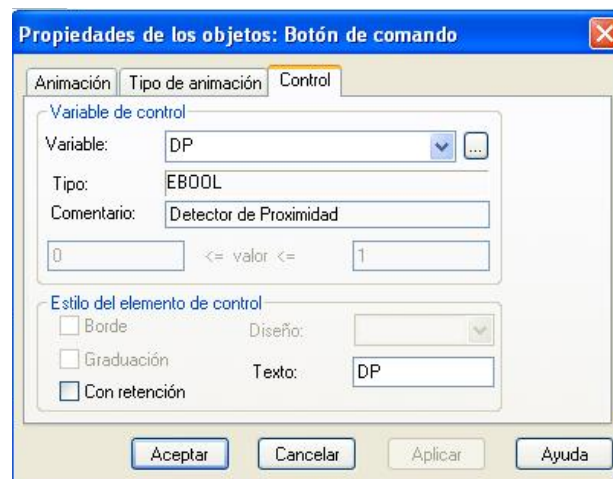


Figura 20: Propiedades de botón de comando

Cinta transportadora:

La cinta transportadora inicialmente muestra el aspecto siguiente, ver Figura 21.



Figura 21: Cinta transportadora

Tal y como hemos dicho, para dar la sensación de movimiento hay que animar las aspas de la cinta. Para ello, se desagrupa la cinta y se accede a cada una de las aspas. En la Figura 22 se muestra la evolución de la cinta y la caja.

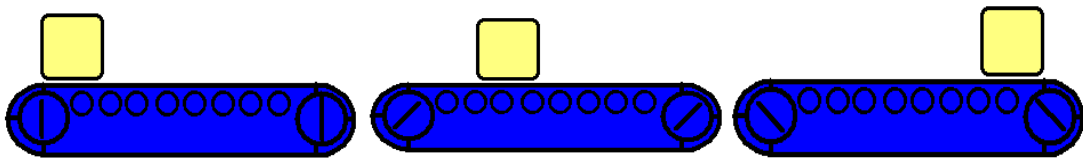


Figura 22: Movimiento de la cinta con la caja

Para configurar cada uno de los elementos, se hace doble clic sobre uno de los objetos, por ejemplo el aspa o la caja, que ha sido dibujada mediante la barra de herramientas, aparece un cuadro de este tipo, ver Figura 23:



Figura 23: Propiedades de una de las aspas de la cinta

Sobre la pestaña “Animación” marcamos la opción Objeto animado para darle movimiento a nuestra presentación. En el campo “variable” se indica la variable a la que hace referencia. Esta será una de las variables creadas para la simulación. Hemos de indicar el tipo de variable que es, e indicamos si es visible todo el tiempo o sólo en función de su valor, 0 o 1. En la pestaña “Tipo de Animación “ elegimos visualización estándar.

Finalmente en la pestaña “Esquema” elegimos otras opciones como los colores, tamaño de la línea, etc.

A continuación se muestra de manera detallada el proceso de simulación de la cinta 1. Tal y como se vio previamente, hay un SFC que controla su evolución, ver Figura 24

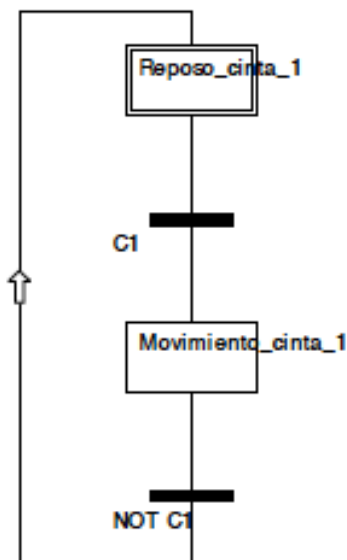
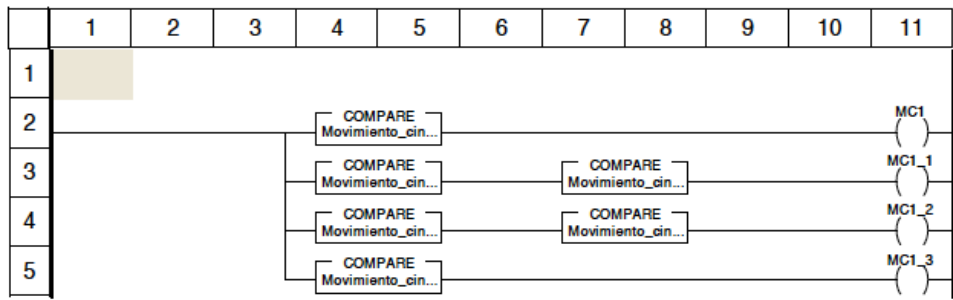


Figura 24: SFC cinta1

Como se observa, la cinta está en reposo hasta que se activa la variable C1 (variable de salida del autómatas), y se desconecta cuando dicha variable se desactiva. Después se activa la etapa de Movimiento_cinta_1. Para programar el movimiento de las aspás se definieron las siguientes variables:

DESCRIPCION	NOMBRE	TIPO	DIRECCIÓN
Movimiento del primer radio de la cinta 1	MC1	BOOL	Sin dirección
Movimiento del segundo radio de la cinta 1	MC1_1	BOOL	Sin dirección
Movimiento del tercer radio de la cinta 1	MC1_2	BOOL	Sin dirección
Movimiento del cuarto radio de la cinta 1	MC1_3	BOOL	Sin dirección

La etapa Movimiento_cinta_1, se programó tal y como se muestra en Figura 25. Las variables se van activando con una evolución temporal, cada segundo.



Etiquetas truncadas:

Etiqueta	Posición(es)
Movimiento cinta 1.t<=t#1s	(4, 2)
Movimiento cinta 1.t<t#2s	(7, 3)
Movimiento cinta 1.t<t#3s	(7, 4)
Movimiento cinta 1.t>=t#1s	(4, 3)
Movimiento cinta 1.t>=t#2s	(4, 4)
Movimiento cinta 1.t>=t#3s	(4, 5)

Figura 25: Etapa Movimiento_cinta_1

El resto de elementos también tienen un SFC asociado activados cada uno por una variable de salida (AR y C2) y en el caso del palet dependerá del número de cajas C. Su programación puede verse en el anexo de este proyecto.

Robot:

En el caso del robot, se ha hecho mediante la unión de varias líneas y agrupándolas finalmente para hacer de todo el conjunto un elemento animado.

Las distintas posiciones del robot se muestran en Figura 26 y cada posición es un objeto animado a su vez y programado de manera similar al aspa de la cinta.

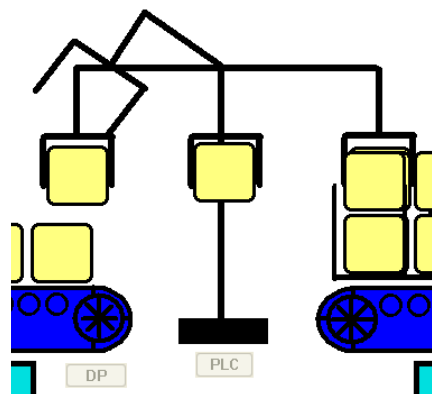


Figura 26: Distintas posiciones de robot

El palet:

Para poder simular el palet se ha decidido dibujar una caja (formada por tres líneas), que permanecerá fija en su posición hasta que se llene, Figura 27.

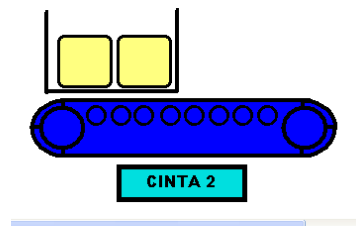


Figura 27: Simulación del palet

Después de eso, el palet irá avanzando junto con la evolución de la cinta 2, tal y como se muestra en Figura 28.

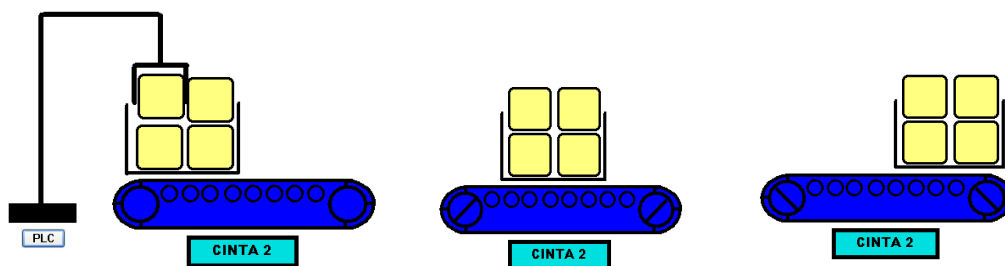


Figura 28: Evolución de la Cinta 2 con el palet (reducida)

4.3 Ejecución del programa

Después de realizar la programación del ejercicio y su pantalla del operador hemos de ejecutar el programa, para ello hemos de compilar el sistema para ver los posibles errores y transferirlo al simulador.

En la barra de herramientas seleccionamos la opción Generar y Regenerar todo el proyecto (ver Figura 29).

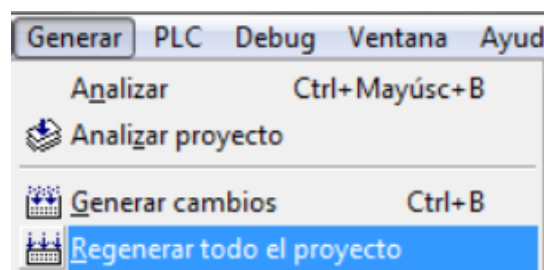


Figura 29: Ventana de regeneración

En el caso de aparecer errores se verán en la pantalla de resultados. Haciendo doble clic sobre la línea el programa nos lleva al lugar donde está el error.

Una vez eliminados los errores, en el caso de que los hubiera, hemos de establecer el modo de simulación en la barra de herramientas, ver Figura 30.

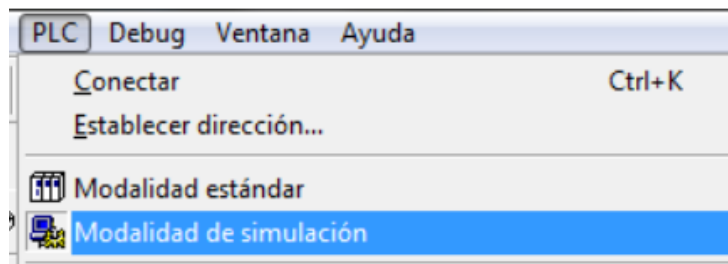


Figura 30: Selección del modo de simulación

Antes de realizar la conexión con el simulador del autómatas hay que establecer la dirección siguiente, tal y como se muestra en la Figura 31:

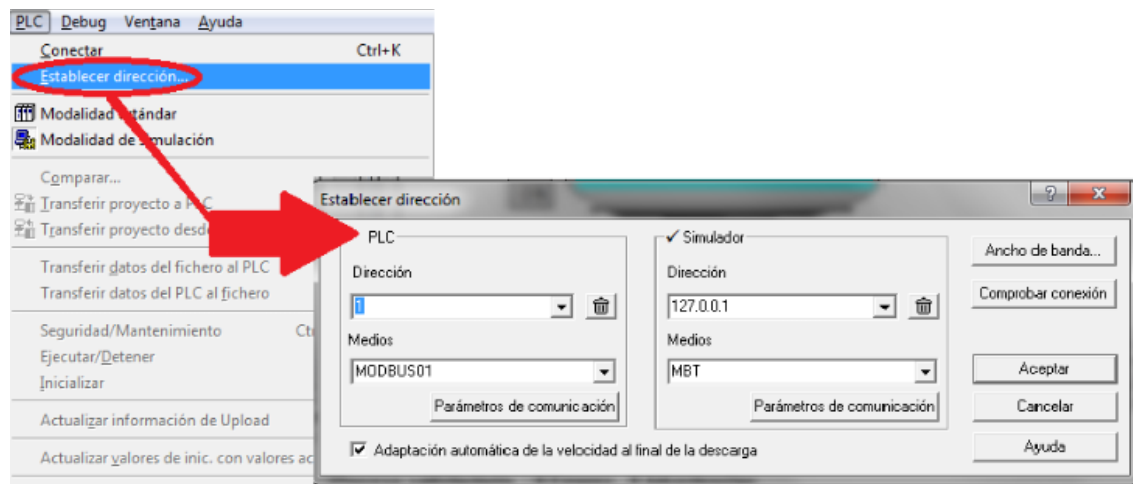


Figura 31: Ventana de establecimiento de dirección del simulador

A continuación, como muestra la Figura 32, podemos conectar con el PLC en modo simulación:

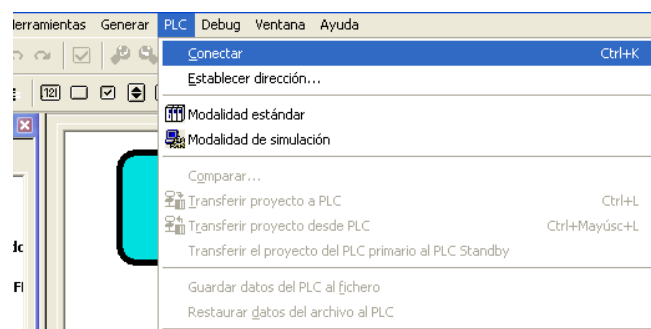


Figura 32: Ventana de conexión al PLC

Posteriormente se transfiere el proyecto al PLC, ver Figura 33.

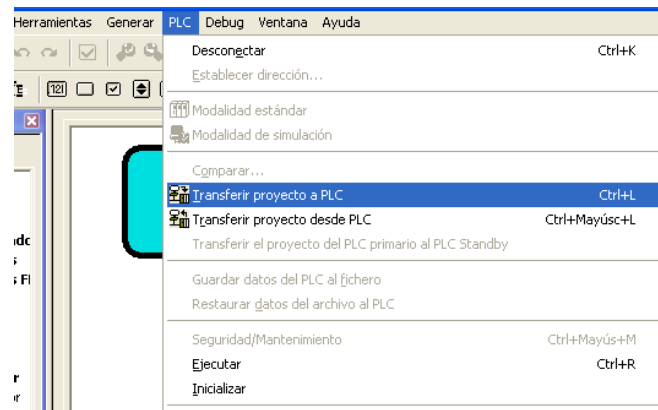


Figura 33: Transferir proyecto

Una vez transferido, solo hay que ejecutar, ver Figura 34, ver :

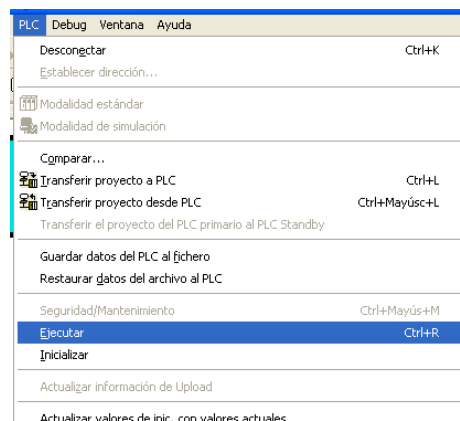


Figura 34: Ejecutar el proyecto

Una vez ejecutado, durante la ejecución, el panel de simulador PLC tendrá el siguiente aspecto, ver Figura 35:



Figura 35: Panel de simulador

Si todo es correcto se enciende en color verde el indicador RUN. Si hay algún error en la configuración aparece encendido en color rojo el indicador ERR.

4.4 Evolución del sistema. Botones de comando

A través de los botones de comando podemos controlar la evolución de la simulación. Para ello, simulamos la activación de las entradas del proceso mediante unos botones que activan dicha señal al pulsar sobre ellos.

Hay que tener especial atención ya que para que funcionen los botones de comando las entradas **no pueden llevar asociada ninguna dirección %I** en la tabla de variables. Otra opción es forzar las señales de entrada en la “Tabla de animación” y ver si funciona nuestra programación.

Es muy importante que durante la simulación el botón “habilitar variables de escritura” este activo, ver Figura 36. Se encuentra en la barra de herramientas. Si no es así, no podremos pulsar los botones.



Figura 36: Habilitar variables de escritura

CAPITULO 5

Modo docente en el laboratorio.

Tal y como se ha explicado en la Motivación de este proyecto, el objetivo es que los alumnos, utilizando las variables imprescindibles, programen correctamente el ejercicio propuesto y la simulación gráfica funcione correctamente.

Por lo tanto, a los alumnos se les daría como material docente:

- El enunciado del ejercicio
- El nombre de las variables que necesitan para programar el funcionamiento del proceso
- El archivo del Unity con la simulación (pantalla de operador y los SFCs de simulación)

La programación del ejercicio se ha hecho teniendo en cuenta que los alumnos puedan usar las pantallas de operador de forma independiente. Por lo tanto, se separó inicialmente la parte de animación de esta pantalla de operador, de la programación del ejercicio.

VARIABLES DEL ALUMNO:

ENTRADAS:

DESCRIPCION	NOMBRE	TIPO	DIRECCIÓN
Interruptor de arranque	IA	EBOOL	Sin dirección
Detector de posición	DP	EBOOL	Sin dirección
Señal robot-PLC	PLC	EBOOL	Sin dirección

SALIDAS:

DESCRIPCION	NOMBRE	TIPO	DIRECCIÓN
Mover cinta 1	C1	EBOOL	%Q.0.2.0
Activar robot	AR	EBOOL	%Q.0.2.1
Mover cinta 2	C2	EBOOL	%Q.0.2.2

OTRAS:

DESCRIPCION	NOMBRE	TIPO	DIRECCIÓN
Contador	C	INT	%MW0

De nuevo, a los alumnos les daremos los SFCs que simulan la evolución gráfica de cada elemento de la pantalla del operador, ver Figura 37. Recordar que las variables que les damos (las variables de simulación) son las que controlan cada uno de los SFCs.

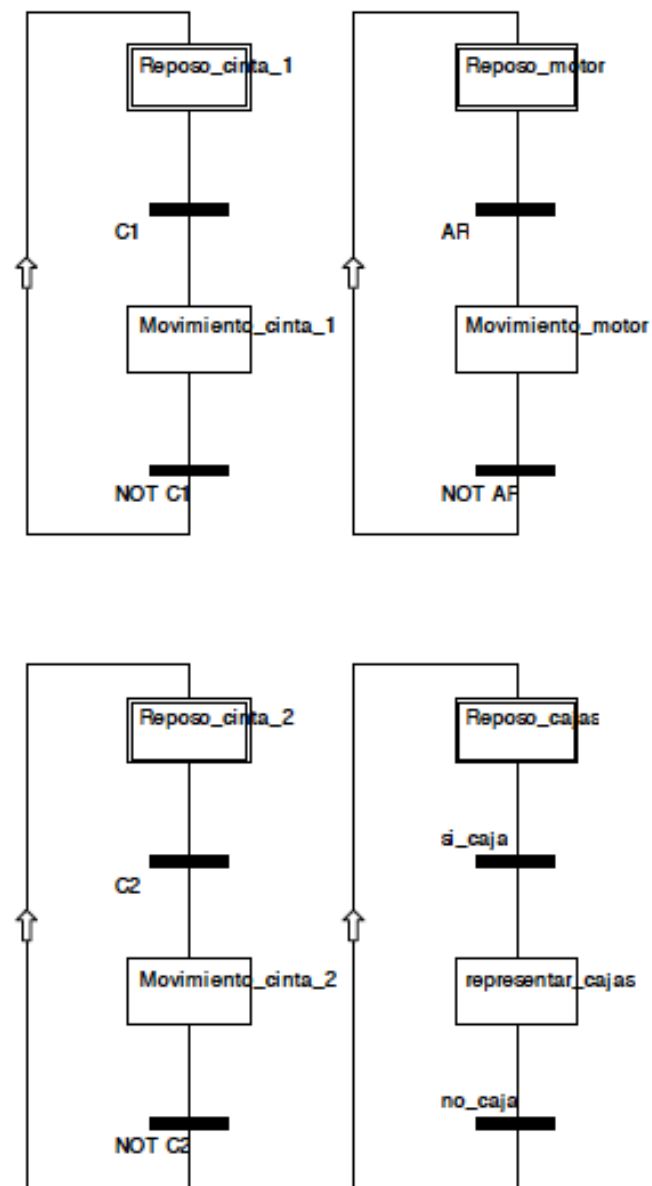


Figura 37: SFCs de los elementos de simulación

VARIABLES CREADAS PARA LA SIMULACION (NO SE LAS DAREMOS A LOS ALUMNOS):

MOVER CINTA 1:

DESCRIPCION	NOMBRE	TIPO	DIRECCIÓN
Movimiento del primer radio de la cinta 1	MC1	BOOL	Sin dirección
Movimiento del segundo radio de la cinta 1	MC1_1	BOOL	Sin dirección
Movimiento del tercer radio de la cinta 1	MC1_2	BOOL	Sin dirección
Movimiento del cuarto radio de la cinta 1	MC1_3	BOOL	Sin dirección

MOVER EL ROBOT:

DESCRIPCION	NOMBRE	TIPO	DIRECCIÓN
Robot en reposo	Motor_0	BOOL	Sin dirección
Robot coge la caja en la cinta 1	Motor_1	BOOL	Sin dirección
Robot girando, en la posición central	Motor_2	BOOL	Sin dirección
Robot deja la caja en la cinta 2	Motor_3	BOOL	Sin dirección

MOVER CINTA 2:

DESCRIPCION	NOMBRE	TIPO	DIRECCIÓN
Movimiento del primer radio de la cinta 2	MC2	BOOL	Sin dirección
Movimiento del segundo radio de la cinta 2	MC2_1	BOOL	Sin dirección
Movimiento del tercer radio de la cinta 2	MC2_2	BOOL	Sin dirección
Movimiento del cuarto radio de la cinta 2	MC2_3	BOOL	Sin dirección

PALET EN MOVIMIENTO:

DESCRIPCION	NOMBRE	TIPO	DIRECCIÓN
Caja 1 en la cinta 2	Caja1	BOOL	Sin dirección
Caja 2 en la cinta 2	Caja2	BOOL	Sin dirección
Caja 3 en la cinta 2	Caja 3	BOOL	Sin dirección
Caja 4 en la cinta 2	Caja4	BOOL	Sin dirección
Palet	Cajon	BOOL	Sin dirección

CAPITULO 6

Bibliografía

- [1] Página web de Scheiner- electric: <http://www.schneider-electric.com/products/es/es/3900-pac-plc-y-otros-controladores/3950-pacs/548-unity-pro/>
(fecha de acceso Octubre 2015)
- [2] INTERNATIONAL ELECTROTECHNICAL COMMISSION. Estandarización en la programación del control industrial. IEC 61131-3. 2003
- [3] MUÑOZ HORCAJUELO, MARÍA. Proyecto Final de Carrera: Simulación de la automatización de procesos con UnityPro. Departamento de Ingeniería de Sistemas y Automática. Universidad Carlos III de Madrid. Año 2012.

CAPITULO 7

Anexos (Documentación Técnica)

DOCUMENTACIÓN TÉCNICA

Proyecto

Proyecto	Proyecto
Diseñador	
Aplicación	DEFINITIVO_MOD.stu
Versión del software	Unity Pro XL V6.0
Fecha de creación	14/10/2015 13:19:59
Fecha de la última modificación	14/10/2015 13:19:59
PLC de destino	TSX P57 104M 02.9057-1, Programa de 224Kb, Unitelway

Autor:	1 Portada	Impreso el 14/10/2015
Dept.:		
Proyecto:		Página: 1 - 1/1

Contenido

1 Portada	1 página
2 Contenido	1 página
3 Configuración	4 páginas
3.1 0 : Bus X	4 páginas
3.1.1 0 : TSX RKY 8EX	4 páginas
3.1.1.1 0 : TSX P57 104M	1 página
3.1.1.2 1 : TSX DEY 16FK	1 página
3.1.1.3 2 : TSX DSY 16R5	1 página
4 Variables e instancias FB	3 páginas
5 Estructura de la aplicación	1 página
6 Programa	19 páginas
6.1 Tareas	19 páginas
6.1.1 MAST	19 páginas
6.1.1.1 Secciones	18 páginas
6.1.1.1.1 Principal	18 páginas
6.1.1.1.1.1 Chart	3 páginas
6.1.1.1.1.2 Acciones	9 páginas
6.1.1.1.1.2.1 IncC	1 página
6.1.1.1.1.2.2 Reposo_Cinta	1 página
6.1.1.1.1.2.3 mov_cinta1	1 página
6.1.1.1.1.2.4 Reposo_Motor_Carga	1 página
6.1.1.1.1.2.5 Mov_motor	1 página
6.1.1.1.1.2.6 Reposo_Cinta_caja	1 página
6.1.1.1.1.2.7 Mov_cinta_2	1 página
6.1.1.1.1.2.8 cajas	1 página
6.1.1.1.1.2.9 cont_0	1 página
6.1.1.1.1.3 Transiciones	5 páginas
6.1.1.1.1.3.1 mayor4	1 página
6.1.1.1.1.3.2 Tiempo5	1 página
6.1.1.1.1.3.3 menor4	1 página
6.1.1.1.1.3.4 si_caja	1 página
6.1.1.1.1.3.5 no_caja	1 página
7 Pantallas de operador	1 página
7.1 Pantalla del operador	1 página
8 Movimiento	1 página
9 Referencias cruzadas	3 páginas

Total: 34 páginas

Autor:	2 Contenido	Impreso el 14/10/2015
Dept.:		
Proyecto:		Página: 2 - 1/1

0 : TSX RKY 8EX

Slot	Familia	Referencia
(P)	Alimentación	TSX PSY 2600M
0	Premium	TSX P57 104M
1	Binario	TSX DEY 16FK
2	Binario	TSX DSY 16R5

Autor:	3.1 0 : Bus X	Impreso el 14/10/2015
Dept.:	3.1.1 0 : TSX RKY 8EX	
Proyecto:		Página: 3.1.1 - 1/4

Este documento es propiedad de XXX y no se puede reproducir ni comercializar sin autorización previa.

0.0 : TSX P57 104M

Identificación del módulo:

Referencia comercial : TSX P57 104M Designación : 57-1, Programa de 224Kb, Unitelway
Dirección : 0.0 Símbolo :

Característica de memoria

Nombre de la tarjeta de memoria : Ninguna

Modalidad de servicio

Entrada de ejecución/detención : No
Protección de memoria : No
Iniciar ejecución automática : No
Resetear MWi : Sí
Sólo arranque en frío : No

Datos

Cantidad de bits : 256
Cantidad de palabras : 512
Cantidad de constantes : 128
Cantidad de bits de sistema : 128
Cantidad de palabras de sistema : 168

Canal 0 :

Función específica de la aplicación : Conexión Uni-Telway
Tipo de canales de E/S : Canal integrado
Tarea : MAST
Tipo : Maestro
Velocidad de transmisión : 19.200 bits/s Datos : a 8 bits
Detener : a 1 bit Paridad : Impar
Cantidad de slaves : 8
Valor de timeout en MS : 30

Canal 100 :

Función específica de la aplicación : Ninguno
Tipo de canales de E/S : Canal integrado

Autor:	3.1.1 0 : TSX RKY 8EX	Impreso el 14/10/2015
Dept.:	3.1.1.1 0 : TSX P57 104M	
Proyecto:		Página: 3.1.1.1 - 1/1

0.1 : TSX DEY 16FK

Identificación del módulo:

Referencia comercial : TSX DEY 16FK
Dirección : 0.1

Designación : 16E RAPID 24 VCC SINK CON.
Símbolo :

Parámetros comunes [0-7]

Monitorización de alimentación : Activo
Tarea : MAST

Parámetros del canal de entrada [0-7]

Canal	Dirección	Símbolo	Filtro	Función
0	%I0.1.0.0		4 ms	
1	%I0.1.1.0		4 ms	
2	%I0.1.2.0		4 ms	
3	%I0.1.3.0		4 ms	
4	%I0.1.4.0		4 ms	
5	%I0.1.5.0		4 ms	
6	%I0.1.6.0		4 ms	
7	%I0.1.7.0		4 ms	

Parámetros comunes [8-15]

Monitorización de alimentación : Activo
Tarea : MAST

Parámetros del canal de entrada [8-15]

Canal	Dirección	Símbolo	Filtro	Función
8			4 ms	
9			4 ms	
10			4 ms	
11			4 ms	
12			4 ms	
13			4 ms	
14			4 ms	
15			4 ms	

0.2 : TSX DSY 16R5

Identificación del módulo:

Referencia comercial : TSX DSY 16R5
Dirección : 0.2

Designación : 16S RELÉ 50 VA, BL.TER.
Símbolo :

Parámetros comunes [0-7]

Tarea : MAST
Modalidad de retorno : Retorno

Parámetros del canal de salida [0-7]

Canal	Dirección	Símbolo	Valor de ret.
0	%Q0.2.0.0	C1	0
1	%Q0.2.1.0	AR	0
2	%Q0.2.2.0	C2	0
3	%Q0.2.3.0		0
4	%Q0.2.4.0		0
5	%Q0.2.5.0		0
6	%Q0.2.6.0		0
7	%Q0.2.7.0		0

Parámetros comunes [8-15]

Tarea : MAST
Modalidad de retorno : Retorno

Parámetros del canal de salida [8-15]

Canal	Dirección	Símbolo	Valor de ret.
8	%Q0.2.8.0		0
9	%Q0.2.9.0		0
10	%Q0.2.10.0		0
11	%Q0.2.11.0		0
12	%Q0.2.12.0		0
13	%Q0.2.13.0		0
14	%Q0.2.14.0		0
15	%Q0.2.15.0		0

Variables e instancias FB

BOOL

Nombre	Const	Dirección	Comentario	Valor	Utilizado	DG
Caja1	NO		Caja 1 en la cinta 2		3	NO
Caja2	NO		Caja 2 en la cinta 2		3	NO
Caja3	NO		Caja 3 en la cinta 2		3	NO
Caja4	NO		Caja 4 en la cinta 2		3	NO
Cajon	NO		Cajón que recoge las cajas		4	NO
mayor4	NO				2	NO
MC1	NO		Movimiento del primer radio de la "rueda" de la cinta 1. Variable creada para la simulacion		14	NO
MC1_1	NO		Movimiento del segundo radio de la "rueda" de la cinta 1. Variable creada para la simulacion		5	NO
MC1_2	NO		Movimiento del tercer radio de la "rueda" de la cinta 1. Variable creada para la simulacion		5	NO
MC1_3	NO		Movimiento del cuarto radio de la "rueda" de la cinta 1. Variable creada para la simulacion		5	NO
MC2	NO		Movimiento del primer radio de la "rueda" de la cinta 2. Variable creada para la simulacion		4	NO
MC2_1	NO		Movimiento del segundo radio de la "rueda" de la cinta 2. Variable creada para la simulacion		4	NO
MC2_2	NO		Movimiento del tercer radio de la "rueda" de la cinta 2. Variable creada para la simulacion		4	NO
MC2_3	NO		Movimiento del cuarto radio de la "rueda" de la cinta 2. Variable creada para la simulacion		5	NO
menor4	NO				2	NO
Motor 0	NO		Motor en estado de reposo		3	NO
Motor 1	NO		Motor cuando coge una caja de la Cinta 1		8	NO
Motor 2	NO		Motor cuando hace la transición hacia la Cinta 2. Movimiento central		6	NO
Motor 3	NO		Motor cuando deja una caja en la Cinta 2		8	NO
no caja	NO				2	NO
si caja	NO				2	NO
Tiempo5	NO				2	NO

EBOOL

Nombre	Const	Dirección	Comentario	Valor	Utilizado	DG
AR	NO	%Q0.2.1	Activar Robot		3	NO
C1	NO	%Q0.2.0	Cinta 1		3	NO
C2	NO	%Q0.2.2	Cinta 2		3	NO
DP	NO		Detector de Proximidad		2	NO
IA	NO		Interruptor de Arranque		2	NO
PLC	NO		Señal del robot al PLC (contador)		2	NO

INT

Nombre	Const	Dirección	Comentario	Valor	Utilizado	DG
C	NO	%MW0	Contador		11	NO

SFCCHART STATE

Nombre	Const	Dirección	Comentario	Utilizado
Principal	NO			0

SFCSTEP STATE

Nombre	Const	Dirección	Comentario	Utilizado
Cinta1	NO		Se activa la Cinta 1	1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			

Autor:	4 Variables e instancias FB	Impreso el 14/10/2015
Dept.:		
Proyecto:		Página: 4 - 1/3

Variables e instancias FB

Nombre	Const	Dirección	Comentario	Utilizado
Cinta2	NO			2
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Contar	NO			1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Movimiento cinta 1	NO			7
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Movimiento cinta 2	NO			8
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Movimiento motor	NO			7
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Reposo	NO		Estado inicial, de reposo. Esperando a que pulsemos el Interruptor de Arranque (IA)	1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Reposo cajas	NO			1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Reposo cinta 1	NO			1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Reposo cinta 2	NO			1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Reposo motor	NO			1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
representar cajas	NO			1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Robot	NO			1
t	NO			
x	NO			

Variables e instancias FB

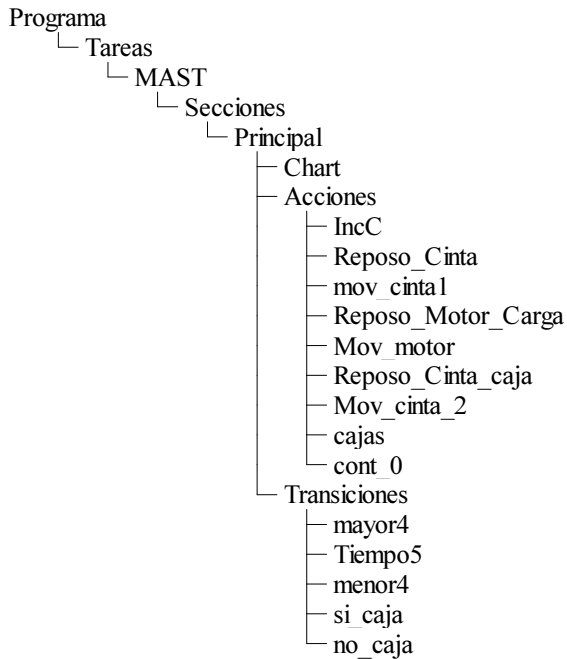
Nombre	Const	Dirección	Comentario	Utilizado
tminErr	NO			
tmaxErr	NO			

Estructura de la aplicación

VISTA ESTRUCTURAL

SECCIÓN	CONDICIÓN DE VALIDACIÓN	COMENTARIO DE SECCIÓN	MÓDULO	LENGUAJE
Principal				SFC
Chart				SFC
IncC				LD
Reposo_Cinta				LD
mov_cinta1				LD
Reposo_Motor_Carga				LD
Mov_motor				LD
Reposo_Cinta_caja				LD
Mov_cinta_2				LD
cajas				LD
cont_0				LD
mayor4				LD
Tiempo5				LD
menor4				LD
si_caja				LD
no_caja				LD

CALL TREE



Autor:	5 Estructura de la aplicación	Impreso el 14/10/2015
Dept.:		
Proyecto:		Página: 5 - 1/1

MAST

Propiedades específicas

Configuración	Cíclica
Configuración del periodo de tareas	0
Configuración del tiempo de watchdog	250

Autor:	6.1 Tareas 6.1.1 MAST	Impreso el 14/10/2015
Dept.:		
Proyecto:		Página: 6.1.1 - 1/19

Principal : [MAST]

Comentario

Propiedades comunes

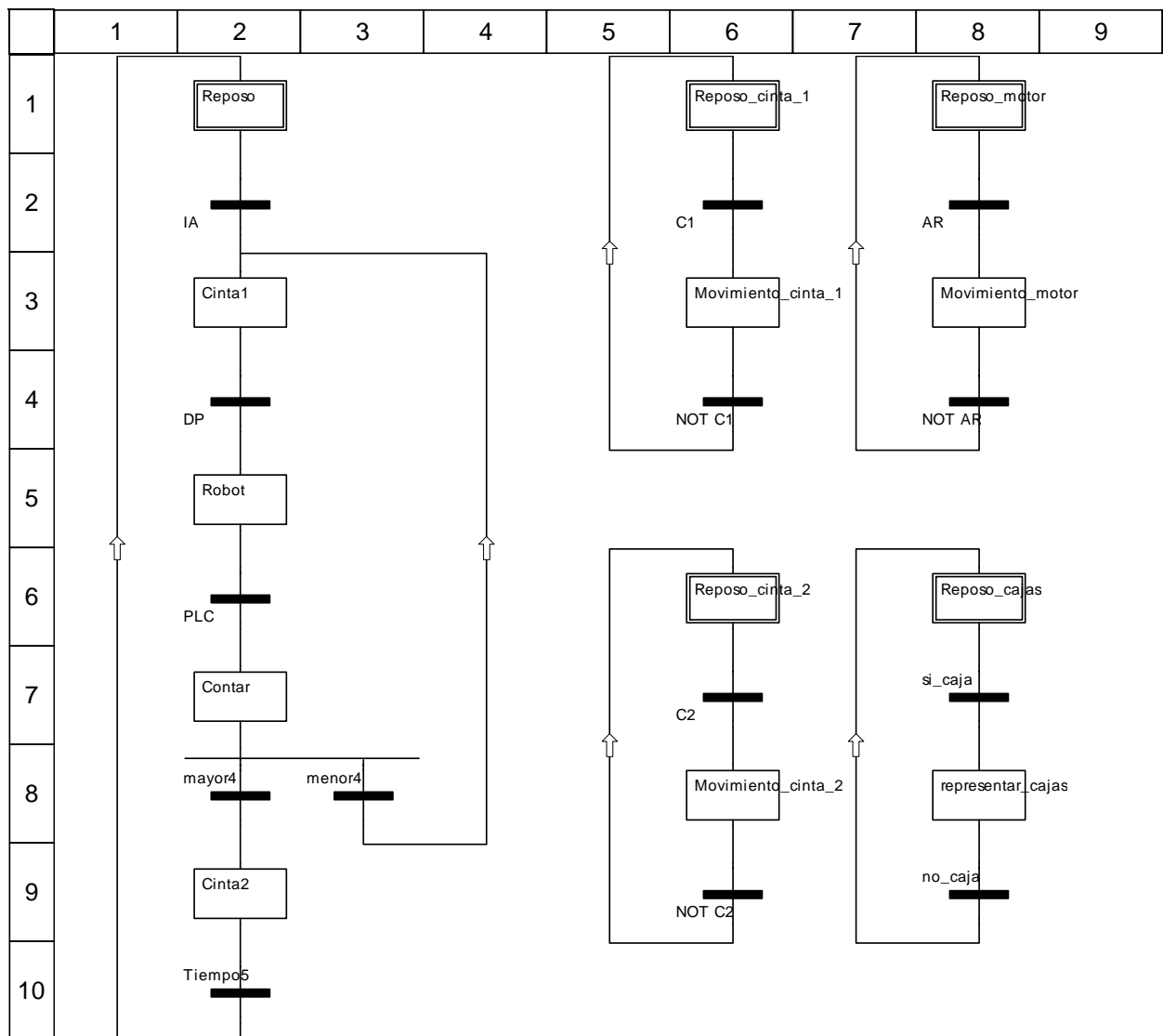
Módulo funcional	
Nombre de la condición	

Propiedades específicas

Control de operador	No
Número de área	0

Autor:	6.1.1.1 Secciones 6.1.1.1.1 Principal	Impreso el 14/10/2015
Dept.:		
Proyecto:		Página: 6.1.1.1.1 - 1/18

Chart : [MAST - Principal]



Descripción de objeto

Pasos:

Cinta1	(2, 3)
Tiempo de supervisión mín./máx.:	Tiempo de retardo:
Comentario: Se activa la Cinta 1	
Acciones:	
Descriptor:	Variable: C1
N	

Cinta2	(2, 9)
Tiempo de supervisión mín./máx.:	Tiempo de retardo:
Comentario:	
Acciones:	

Autor:	6.1.1.1.1 Principal	Impreso el 14/10/2015
Dept.:	6.1.1.1.1 Chart	
Proyecto:		Página: 6.1.1.1.1 - 1/3

Descriptor: N	Tiempo:	Variable: C2
Descriptor: P1	Tiempo:	Sección: LD :: cont_0

Contar		(2, 7)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: P1	Tiempo:	Sección: LD :: IncC

Movimiento cinta 1		(6, 3)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Sección: LD :: mov_cinta1

Movimiento cinta 2		(6, 8)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Sección: LD :: Mov_cinta_2

Movimiento motor		(8, 3)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Sección: LD :: Mov_motor

Reposo (paso inicial)	(2, 1)
Tiempo de supervisión mín./máx.:	Tiempo de retardo:
Comentario:	Estado inicial, de reposo. Esperando a que pulsemos el Interruptor de Arranque (IA)

Reposo cajas (paso inicial)	(8, 6)
Tiempo de supervisión mín./máx.:	Tiempo de retardo:
Comentario:	

Reposo cinta 1 (paso inicial)		(6, 1)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: None	Tiempo:	Sección: LD :: Reposo_Cinta

Reposo cinta 2 (paso inicial)		(6, 6)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Sección: LD :: Reposo_Cinta_caja
Descriptor: N	Tiempo:	Variable: Cajon

Reposo motor (paso inicial)		(8, 1)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: None	Tiempo:	Sección: LD :: Reposo_Motor_Carga

Autor:	6.1.1.1.1 Principal	Impreso el 14/10/2015
Dept.:	6.1.1.1.1 Chart	
Proyecto:		Página: 6.1.1.1.1 - 2/3

Robot			(2, 5)
Tiempo de supervisión mín./máx.:			Tiempo de retardo:
Comentario:			
Acciones:			
Descriptor: N	Tiempo:	Variable: AR	

representar cajas		(8, 8)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Sección: LD :: cajas

Transiciones:

Nombre	Tipo de condición	Posición	Comentario
AR	Variable	(8, 2)	Activo el motor
C1	Variable	(6, 2)	
C2	Variable	(6, 7)	
DP	Variable	(2, 4)	Cuando llega al final de la cinta se activa el Detector de Proximidad (DP)
IA	Variable	(2, 2)	Se activa cuando pulsamos el Interruptor de Arranque (IA)
NOT AR	Variable	(8, 4)	
NOT C1	Variable	(6, 4)	
NOT C2	Variable	(6, 9)	
PLC	Variable	(2, 6)	Señal del Robot, el automata así puede ir comparando y ver cuantas cajas hay
LD :: Tiempo5	Sección	(2, 10)	La cinta2 se activa durante 5 segundos
LD :: mayor4	Sección	(2, 8)	
LD :: menor4	Sección	(3, 8)	
LD :: no_caja	Sección	(8, 9)	
LD :: si_caja	Sección	(8, 7)	

IncC <Acción> : [MAST - Principal]

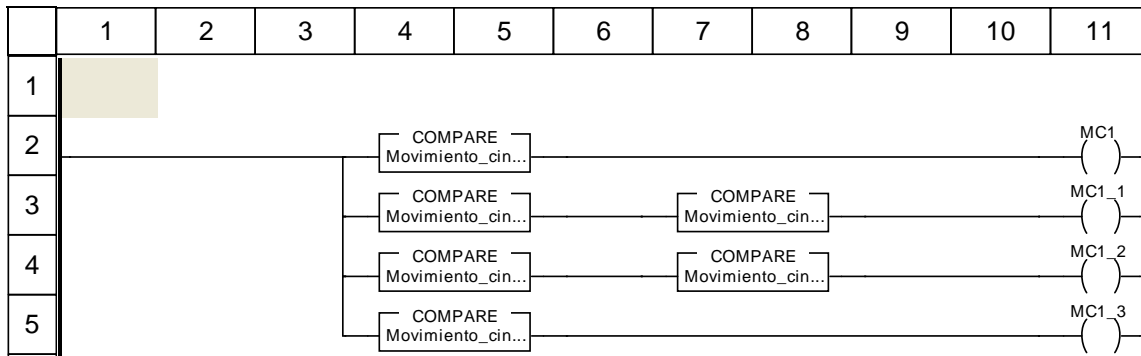
	1	2	3	4	5	6	7	8	9	10	11
1											
2											

OPERATE
Inc(C);

Reposo_Cinta <Acción> : [MAST - Principal]

	1	2	3	4	5	6	7	8	9	10	11
1											
2										MC1 (/)	
3										MC1_1 (/)	
4										MC1_2 (/)	
5										MC1_3 (/)	

mov_cinta1 <Acción> : [MAST - Principal]



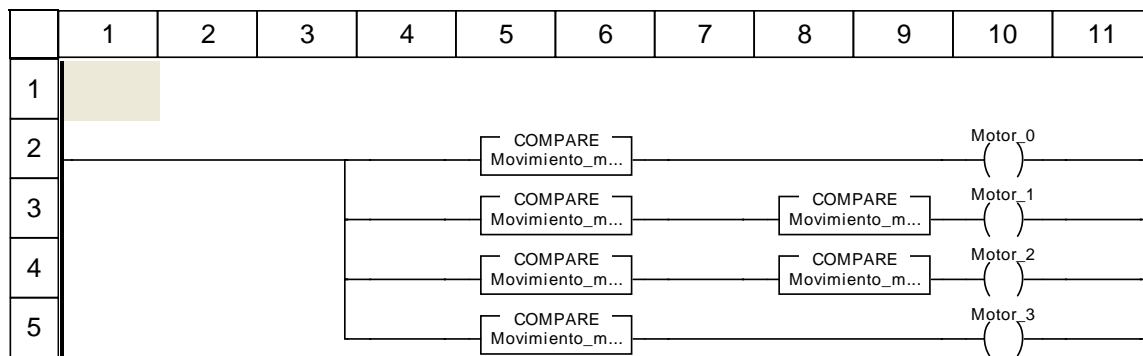
Etiquetas truncadas:

Etiqueta	Posición(es)
Movimiento cinta 1.t<=t#1s	(4, 2)
Movimiento cinta 1.t<t#2s	(7, 3)
Movimiento cinta 1.t<t#3s	(7, 4)
Movimiento cinta 1.t>=t#1s	(4, 3)
Movimiento cinta 1.t>=t#2s	(4, 4)
Movimiento cinta 1.t>=t#3s	(4, 5)

Reposo_Motor_Carga <Acción> : [MAST - Principal]

	1	2	3	4	5	6	7	8	9	10	11
1											
2											Motor_0 (/)
3											Motor_1 (/)
4											Motor_2 (/)
5											Motor_3 (/)

Mov_motor <Acción> : [MAST - Principal]



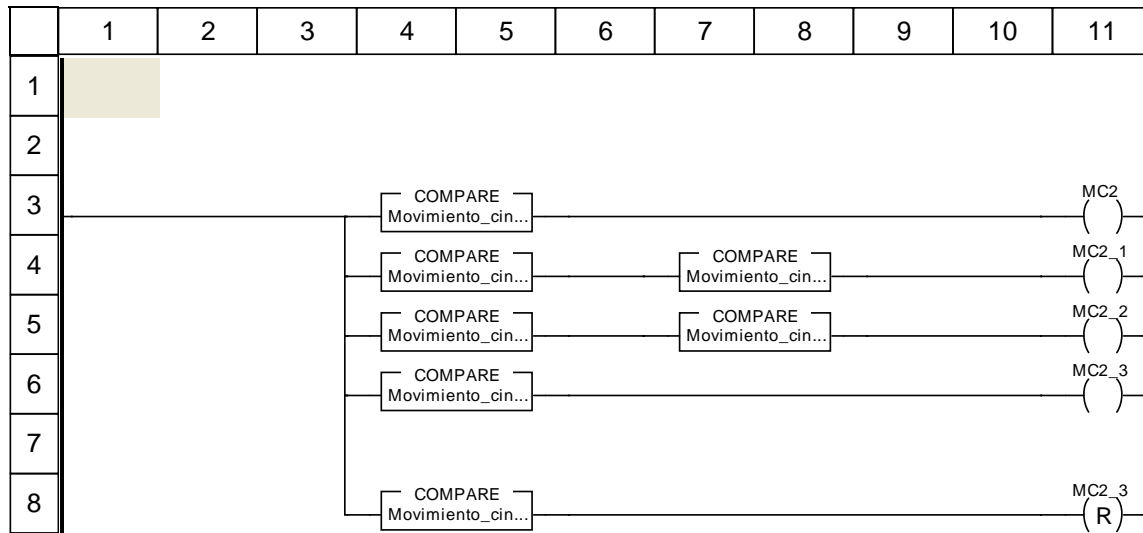
Etiquetas truncadas:

Etiqueta	Posición(es)
Movimiento motor.t<t#1s	(5, 2)
Movimiento motor.t<t#2s	(8, 3)
Movimiento motor.t<t#3s	(8, 4)
Movimiento motor.t>=t#1s	(5, 3)
Movimiento motor.t>=t#2s	(5, 4)
Movimiento motor.t>=t#3s	(5, 5)

Reposo_Cinta_caja <Acción> : [MAST - Principal]

	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											Cajon ()

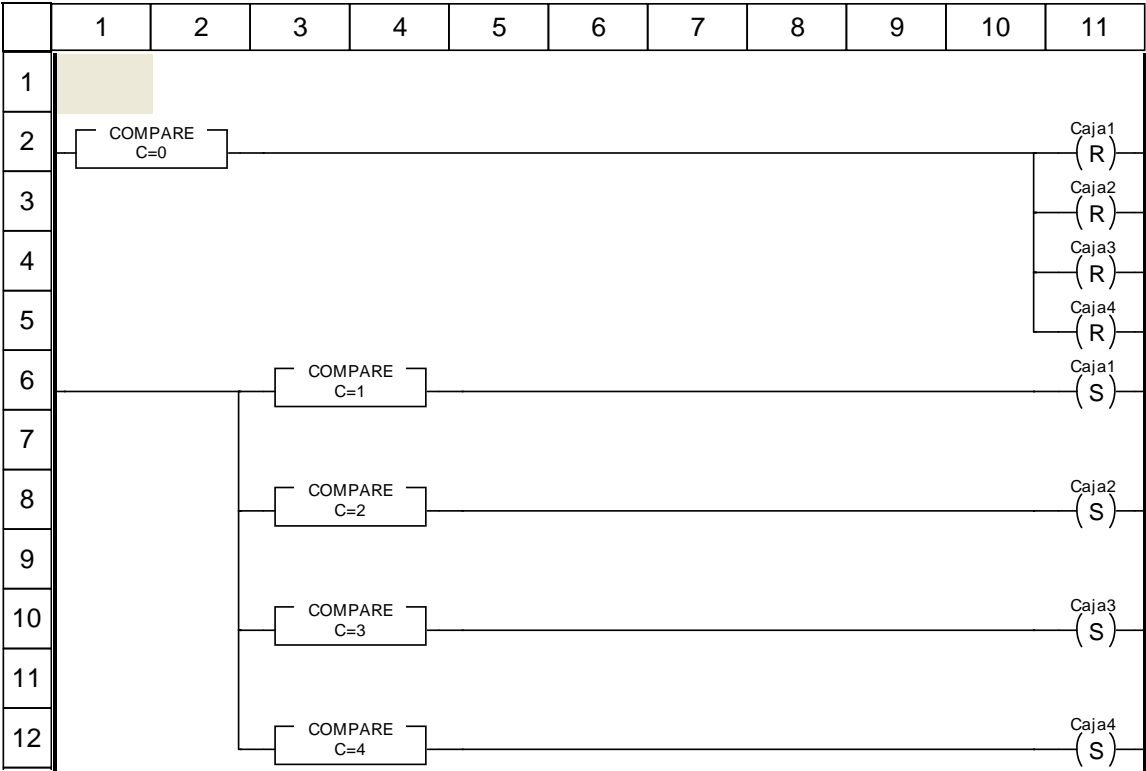
Mov_cinta_2 <Acción> : [MAST - Principal]



Etiquetas truncadas:

Etiqueta	Posición(es)
Movimiento cinta 2.t<=t#1s	(4, 3)
Movimiento cinta 2.t<t#2s	(7, 4)
Movimiento cinta 2.t<t#3s	(7, 5)
Movimiento cinta 2.t>=t#1s	(4, 4)
Movimiento cinta 2.t>=t#2s	(4, 5)
Movimiento cinta 2.t>=t#3s	(4, 6)
Movimiento cinta 2.t>=t#5s	(4, 8)

cajas <Acción> : [MAST - Principal]

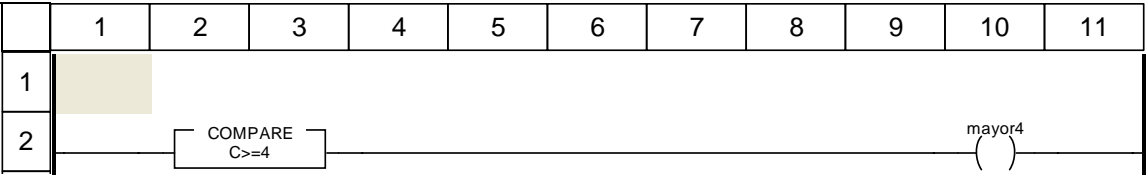


cont_0 <Acción> : [MAST - Principal]

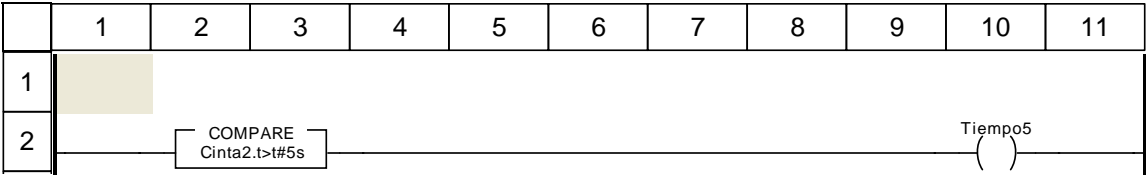
	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											

OPERATE
C:=0;

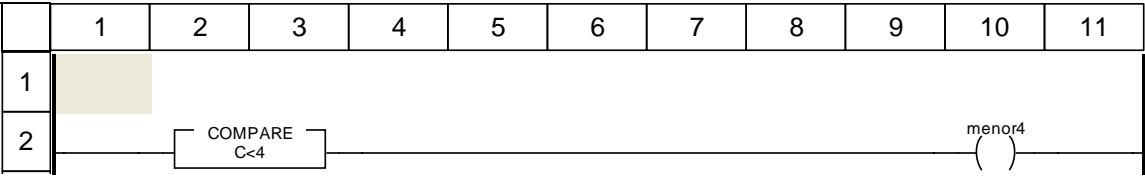
mayor4 <Transición> : [MAST - Principal]



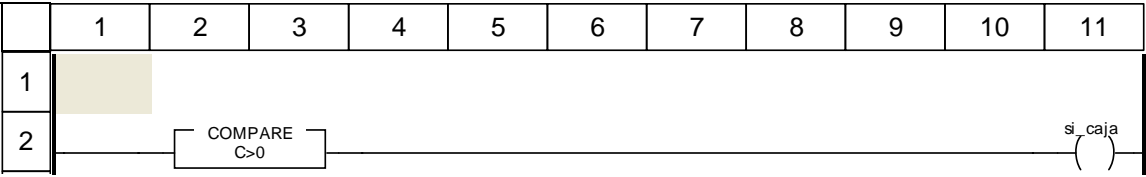
Tiempo5 <Transición> : [MAST - Principal]



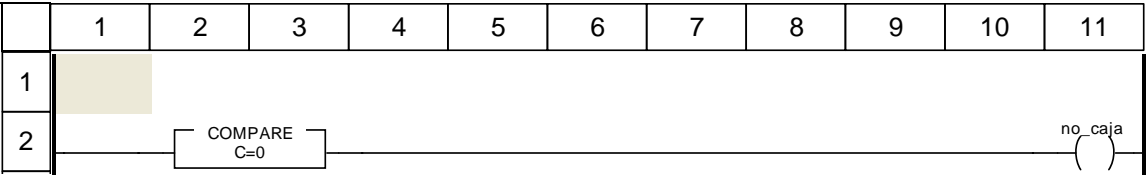
menor4 <Transición> : [MAST - Principal]



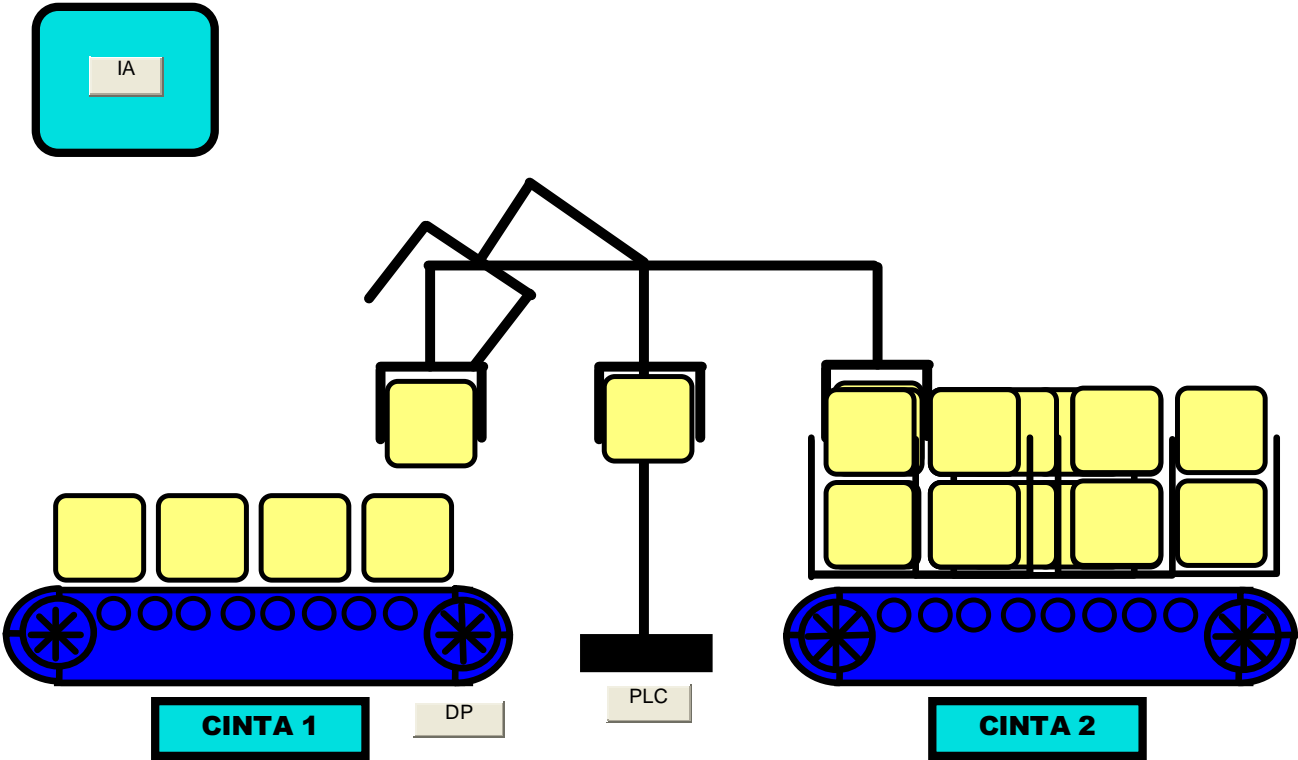
si_caja <Transición> : [MAST - Principal]



no_caja <Transición> : [MAST - Principal]



Pantalla del operador



Eje de movimiento

Autor:	8 Movimiento	Impreso el 14/10/2015
Dept.:		
Proyecto:		Página: 8 - 1/1

Este documento es propiedad de XXX y no se puede reproducir ni comercializar sin autorización previa.

Referencias cruzadas

Aplicación:

Direcciones

Objeto	Referenciado dentro	Etiqueta	Uso
--------	---------------------	----------	-----

Variables o instancias FB

Objeto	Referenciado dentro	Etiqueta	Uso
AR	Chart : [MAST - Principal]	(l: 5, c: 2)	E
		(l: 2, c: 8)	L
		(l: 4, c: 8)	L
C	mayor4 <Transición> : [MAST - Principal]	(l: 2, c: 2)	L
	cajas <Acción> : [MAST - Principal]	(l: 2, c: 1)	L
		(l: 8, c: 3)	L
		(l: 10, c: 3)	L
		(l: 12, c: 3)	L
		(l: 6, c: 3)	L
	IncC <Acción> : [MAST - Principal]	(l: 2, c: 8)	L/E
	no_caja <Transición> : [MAST - Principal]	(l: 2, c: 2)	L
	si_caja <Transición> : [MAST - Principal]	(l: 2, c: 2)	L
	cont_0 <Acción> : [MAST - Principal]	(l: 3, c: 8)	E
	menor4 <Transición> : [MAST - Principal]	(l: 2, c: 2)	L
C1	Chart : [MAST - Principal]	(l: 2, c: 6)	L
		(l: 4, c: 6)	L
		(l: 3, c: 2)	E
C2	Chart : [MAST - Principal]	(l: 9, c: 2)	E
		(l: 7, c: 6)	L
		(l: 9, c: 6)	L
Caja1	cajas <Acción> : [MAST - Principal]	(l: 2, c: 11)	E
		(l: 6, c: 11)	E
	<Pantalla>Pantalla del operador		L (x1)
			INDEF (x4)
Caja2	<Pantalla>Pantalla del operador		L (x1)
	cajas <Acción> : [MAST - Principal]	(l: 8, c: 11)	E
		(l: 3, c: 11)	E
Caja3	cajas <Acción> : [MAST - Principal]	(l: 10, c: 11)	E
		(l: 4, c: 11)	E
	<Pantalla>Pantalla del operador		L (x1)
			INDEF (x4)
Caja4	<Pantalla>Pantalla del operador		L (x1)
	cajas <Acción> : [MAST - Principal]	(l: 12, c: 11)	E
		(l: 5, c: 11)	E
Cajon	Chart : [MAST - Principal]	(l: 6, c: 6)	E
	<Pantalla>Pantalla del operador		L (x2)
			INDEF (x5)
	Reposo_Cinta_caja <Acción> : [MAST - Principal]	(l: 3, c: 11)	E
Cinta1	Chart : [MAST - Principal]	(l: 3, c: 2)	E
		(l: 8, c: 3)	REF E
Cinta2	Chart : [MAST - Principal]	(l: 9, c: 2)	E
	Tiempo5 <Transición> : [MAST - Principal]	(l: 2, c: 2)	L
Contar	Chart : [MAST - Principal]	(l: 7, c: 2)	E
DP	<Pantalla>Pantalla del operador		L/E (x1)
			INDEF (x1)
	Chart : [MAST - Principal]	(l: 4, c: 2)	L
IA	<Pantalla>Pantalla del operador		L/E (x1)
	Chart : [MAST - Principal]	(l: 2, c: 2)	L
MC1	Reposo_Cinta <Acción> : [MAST - Principal]	(l: 2, c: 10)	E

Autor:	9 Referencias cruzadas	Impreso el 14/10/2015
Dept.:		
Proyecto:		Página: 9 - 1/3

Este documento es propiedad de XXX y no se puede reproducir ni comercializar sin autorización previa.

Referencias cruzadas

Objeto	Referenciado dentro	Etiqueta	Uso
	mov_cinta1 <Acción> : [MAST - Principal]	(l: 2, c: 11)	E
	<Pantalla>Pantalla del operador		L (x12)
			INDEF (x17)
MC1_1	mov_cinta1 <Acción> : [MAST - Principal]	(l: 3, c: 11)	E
	<Pantalla>Pantalla del operador		L (x3)
			INDEF (x6)
	Reposo_Cinta <Acción> : [MAST - Principal]	(l: 3, c: 10)	E
MC1_2	mov_cinta1 <Acción> : [MAST - Principal]	(l: 4, c: 11)	E
	<Pantalla>Pantalla del operador		L (x3)
			INDEF (x6)
	Reposo_Cinta <Acción> : [MAST - Principal]	(l: 4, c: 10)	E
MC1_3	mov_cinta1 <Acción> : [MAST - Principal]	(l: 5, c: 11)	E
	<Pantalla>Pantalla del operador		L (x3)
			INDEF (x10)
	Reposo_Cinta <Acción> : [MAST - Principal]	(l: 5, c: 10)	E
MC2	<Pantalla>Pantalla del operador		L (x3)
			INDEF (x8)
	Mov_cinta_2 <Acción> : [MAST - Principal]	(l: 3, c: 11)	E
MC2_1	<Pantalla>Pantalla del operador		L (x3)
			INDEF (x6)
	Mov_cinta_2 <Acción> : [MAST - Principal]	(l: 4, c: 11)	E
MC2_2	<Pantalla>Pantalla del operador		L (x3)
			INDEF (x6)
	Mov_cinta_2 <Acción> : [MAST - Principal]	(l: 5, c: 11)	E
MC2_3	<Pantalla>Pantalla del operador		L (x3)
			INDEF (x3)
	Mov_cinta_2 <Acción> : [MAST - Principal]	(l: 6, c: 11)	E
		(l: 8, c: 11)	E
Motor_0	Mov_motor <Acción> : [MAST - Principal]	(l: 2, c: 10)	E
	Reposo_Motor_Carga <Acción> : [MAST - Principal]	(l: 2, c: 11)	E
	<Pantalla>Pantalla del operador		L (x1)
Motor_1	Mov_motor <Acción> : [MAST - Principal]	(l: 3, c: 10)	E
	Reposo_Motor_Carga <Acción> : [MAST - Principal]	(l: 3, c: 11)	E
	<Pantalla>Pantalla del operador		L (x6)
			INDEF (x7)
Motor_2	Mov_motor <Acción> : [MAST - Principal]	(l: 4, c: 10)	E
	Reposo_Motor_Carga <Acción> : [MAST - Principal]	(l: 4, c: 11)	E
	<Pantalla>Pantalla del operador		L (x4)
Motor_3	Mov_motor <Acción> : [MAST - Principal]	(l: 5, c: 10)	E
	Reposo_Motor_Carga <Acción> : [MAST - Principal]	(l: 5, c: 11)	E
	<Pantalla>Pantalla del operador		L (x6)
Movimiento_cinta_1	mov_cinta1 <Acción> : [MAST - Principal]	(l: 2, c: 4)	L
		(l: 3, c: 4)	L
		(l: 3, c: 7)	L
		(l: 4, c: 4)	L
		(l: 4, c: 7)	L
		(l: 5, c: 4)	L
	Chart : [MAST - Principal]	(l: 3, c: 6)	E
Movimiento_cinta_2	Chart : [MAST - Principal]	(l: 8, c: 6)	E
	Mov_cinta_2 <Acción> : [MAST - Principal]	(l: 4, c: 7)	L
		(l: 5, c: 7)	L
		(l: 6, c: 4)	L
		(l: 5, c: 4)	L

Referencias cruzadas

Objeto	Referenciado dentro	Etiqueta	Uso
		(l: 8, c: 4)	L
		(l: 3, c: 4)	L
		(l: 4, c: 4)	L
Movimiento motor	Mov motor <Acción> : [MAST - Principal]	(l: 2, c: 5)	L
		(l: 3, c: 5)	L
		(l: 3, c: 8)	L
		(l: 4, c: 5)	L
		(l: 4, c: 8)	L
		(l: 5, c: 5)	L
	Chart : [MAST - Principal]	(l: 3, c: 8)	E
PLC	<Pantalla>Pantalla del operador		L/E (x1)
	Chart : [MAST - Principal]	(l: 6, c: 2)	L
Reposo	Chart : [MAST - Principal]	(l: 1, c: 2)	E
		(l: 10, c: 2)	REF E
Reposo cajas	Chart : [MAST - Principal]	(l: 6, c: 8)	E
		(l: 9, c: 8)	REF E
Reposo cinta 1	Chart : [MAST - Principal]	(l: 1, c: 6)	E
		(l: 4, c: 6)	REF E
Reposo cinta 2	Chart : [MAST - Principal]	(l: 6, c: 6)	E
		(l: 9, c: 6)	REF E
Reposo motor	Chart : [MAST - Principal]	(l: 1, c: 8)	E
		(l: 4, c: 8)	REF E
Robot	Chart : [MAST - Principal]	(l: 5, c: 2)	E
Tiempo5	Chart : [MAST - Principal]	(l: 10, c: 2)	L
	Tiempo5 <Transición> : [MAST - Principal]	(l: 2, c: 10)	E
mayor4	Chart : [MAST - Principal]	(l: 8, c: 2)	L
	mayor4 <Transición> : [MAST - Principal]	(l: 2, c: 10)	E
menor4	menor4 <Transición> : [MAST - Principal]	(l: 2, c: 10)	E
	Chart : [MAST - Principal]	(l: 8, c: 3)	L
no_caja	Chart : [MAST - Principal]	(l: 9, c: 8)	L
	no_caja <Transición> : [MAST - Principal]	(l: 2, c: 11)	E
representar_cajas	Chart : [MAST - Principal]	(l: 8, c: 8)	E
si_caja	Chart : [MAST - Principal]	(l: 7, c: 8)	L
	si_caja <Transición> : [MAST - Principal]	(l: 2, c: 11)	E

Objetos EF

Objeto	Referenciado dentro	Etiqueta	Uso
inc	IncC <Acción> : [MAST - Principal]	(l: 2, c: 8)	LLAM F
inc_int	IncC <Acción> : [MAST - Principal]	(l: 2, c: 8)	LLAM F

Subrutinas

Objeto	Referenciado dentro	Etiqueta	Uso
--------	---------------------	----------	-----